

UNIVERSIDADE FEDERAL DO PARANÁ

ALEXANDRE VELOSO DE MATOS

UMA ARQUITETURA PARA A GERÊNCIA DA MIGRAÇÃO DE  
MÁQUINAS VIRTUAIS EM AMBIENTES DE COMPUTAÇÃO EM NUVEM

CURITIBA PR

2020

ALEXANDRE VELOSO DE MATOS

UMA ARQUITETURA PARA A GERÊNCIA DA MIGRAÇÃO DE  
MÁQUINAS VIRTUAIS EM AMBIENTES DE COMPUTAÇÃO EM NUVEM

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Carlos Alberto Maziero.

CURITIBA PR

2020

Catálogo na Fonte: Sistema de Bibliotecas, UFPR  
Biblioteca de Ciência e Tecnologia

M433a

Matos, Alexandre Veloso de

Uma arquitetura para a gerência da migração de máquinas virtuais em ambientes de computação em nuvem [recurso eletrônico] / Alexandre Veloso de Matos. – Curitiba, 2020.

Tese - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2020.

Orientador: Carlos Alberto Maziero

1. Computação em nuvem. 2. Processamento eletrônico de dados – Processamento distribuído. 3. Armazenamento virtual (Ciência da computação). I. Universidade Federal do Paraná. II. Maziero, Carlos Alberto. III. Título.

CDD: 004.6782

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **ALEXANDRE VELOSO DE MATOS** intitulada: **Uma Arquitetura para a Gerência da Migração de Máquinas Virtuais em Ambientes de Computação em Nuvem**, sob orientação do Prof. Dr. CARLOS ALBERTO MAZIERO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 29 de Outubro de 2020.

Assinatura Eletrônica  
30/10/2020 10:48:05.0  
CARLOS ALBERTO MAZIERO  
Presidente da Banca Examinadora

Assinatura Eletrônica  
03/11/2020 14:09:37.0  
ALCIDES CALSAVARA  
Avaliador Externo (PONTIFICA UNIVERSIDADE CATÓLICA DO  
PARANÁ)

Assinatura Eletrônica  
30/10/2020 10:48:13.0  
ELIAS PROCÓPIO DUARTE JUNIOR  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica  
30/10/2020 14:50:34.0  
LUCIANO PASCHOAL GASPARY  
Avaliador Externo (UNIVERSIDADE FEDERAL DO RIO GRANDE DO  
SUL)

*Seria injusto dedicar este trabalho a apenas uma pessoa. Contudo, personifico em todos os familiares, amigos e colegas a figura de minha mãe, a Dona Diléi. Ao dedicar esse trabalho a ela, espero que cada um dos que estiveram (ainda estão e com certeza sempre estarão) vibrando pela minha conquista, consigam entender que também lhes dedico este trabalho.*

## **AGRADECIMENTOS**

Apesar de um trabalho de doutorado ser considerado uma espécie de viagem solitária, eu pude, ao longo do trajeto dessa viagem, constatar que não é. Muitas pessoas foram (e ainda são) essenciais na minha formação como pesquisador, por isso, a todas elas eu envio meu agradecimento. Agradeço ao meu orientador, o Prof. Carlos Maziero, ou o estimado Maziero, a quem dedico minha sincera admiração e respeito. Agradeço à minha família que, durante tantos anos, me acompanhou em outras viagens e que carinhosamente me incentivou a não desistir - a eles, meu pai Hélio, meus irmãos Aurélio, Olímpia, Luciano e Lúcio, amorosamente, dedico um agradecimento especial. Agradeço aos muitos amigos que fizeram desta viagem um roteiro de inúmeros aprendizados, de alegrias, de coragem e de amor. Seria injusto enumerar alguns nomes, pois, com certeza meu cérebro me trairia, pois sei que ele não tem a capacidade de abarcar a gigante extensão de boas almas que me elegeram como amigo. Contudo, tenho em cada um dos meus amigos que um dia lerão esses agradecimentos a minha imensa gratidão. A todos que fizeram com que eu me conhecesse de fato e percebesse que há em mim, ao contrário do que inicialmente eu imaginava, um potencial, só tenho que dedicar o meu eterno carinho. Enfim, agradeço a todos pela valorosa viagem e de ter encontrado em mim o que sempre achei que não existia: a capacidade de ser um doutor.

## RESUMO

A virtualização oferece uma relevante contribuição para a consolidação dos propósitos das nuvens computacionais. Serviços em Nuvem (*Cloud Services* - CSs) surgiram com a perspectiva de atender à crescente variedade de perfis de consumidores. Nesse contexto, as Máquinas Virtuais (VMs) possuem um papel central, dada a flexibilidade decorrente do mapeamento de recursos físicos em virtuais. No entanto, quanto mais disseminada a utilização de CSs, maior é a tendência da sobrecarga dos recursos. Em decorrência, episódios de não cumprimento dos objetivos traçados em acordos (SLAs) tendem a ocorrer com maior frequência. Ações corretivas apenas encontram alguma solução quando associadas a esforços de gerência dos recursos, tais como a migração de VMs. Os hipervisores são essenciais no planejamento e na execução das migrações. Entretanto, embora promovam a garantia do isolamento de aplicações e, por consequência no controle do uso de recursos, seu espectro de ação é limitado. Capacidades de gerência que integrem e coordenem o uso de recursos ainda são incipientes. É, portanto, relevante considerar uma solução que potencialize as contribuições da migração de VMs. Tal solução requer uma orquestração entre os distintos níveis de decisão envolvidos em uma ação de migração: no nível operacional, a ação dos *hipervisores* precisa ser coordenada; no nível decisório, decisões necessitam ser guiadas pelo suporte estratégico de políticas que são delineadas em um nível de controle e gerência da migração. Esta tese apresenta uma solução para a gerência da migração de VMs, destacando um modelo de arquitetura que possibilita atingir tal propósito. Simulações possibilitaram a avaliação da arquitetura, destacando que a adoção das ações de gerência da arquitetura tornam os resultados de uma migração de VMs mais perenes, diminuindo novas instabilidades que, usualmente, indicariam a necessidade de novas migrações.

Palavras-chave: Nuvens computacionais, Migração de máquinas virtuais, Máquinas Virtuais.

## **ABSTRACT**

Virtualization offers a relevant contribution to the realm of Cloud Computing. Cloud Services (CSs) are a relevant approach, aiming to achieve several consumer profiles. Within such context, Virtual Machines (VMs) play a core role, regarding their flexibility due the easiness for mapping a physical resource in a virtual correspondent. However, as CSs are instantiated, the larger is the amount of resources required. The correct of issues can be successful if they are associated with resource management actions, such as the migration of VMs. However, the migration demands management efforts that are complex. Although hypervisors play an important role for the management, its actions are limited. In fact, there is still a lack of management actions that enable the integration and coordination of resource usage in cloud domains. As a consequence, it is important to consider a solution that enhances the contribution of the migration of VMs in the management of cloud resources. Such a contribution demands the coordination of distinct decision levels involved in a migration. In the operational level, it is necessary to coordinate distinct hypervisors; for the decision level, the support of both strategical and technical information are crucial to present a comprehensive approach to manage migrations. This thesis presents one solution to the management of VM migrations, proposing an architecture that meet that purpose. Simulations were executed to evaluate the architecture, highlighting that the adoption of management actions within the architecture aid in the stabilization of migration results, e.g., reducing that new instabilities arose as well as new migration needs.

**Keywords:** Clouds, Migration of virtual machines, Virtual Machines.



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>9</b>
1.1	CONTEXTUALIZAÇÃO . . . . .	9
1.2	O PROBLEMA . . . . .	9
1.3	OBJETIVOS . . . . .	10
1.4	CONTRIBUIÇÕES . . . . .	11
1.5	ORGANIZAÇÃO DO DOCUMENTO . . . . .	11
<b>2</b>	<b>CONCEITOS DE MÁQUINAS VIRTUAIS . . . . .</b>	<b>12</b>
2.1	MÁQUINAS VIRTUAIS E HIPERVISORES . . . . .	12
2.2	A MIGRAÇÃO DE VMS . . . . .	14
2.3	CUSTOS DE UMA MIGRAÇÃO . . . . .	18
2.4	CICLO DE VIDA DE UMA MIGRAÇÃO . . . . .	20
2.5	CONSIDERAÇÕES FINAIS . . . . .	22
<b>3</b>	<b>NUVENS COMPUTACIONAIS . . . . .</b>	<b>24</b>
3.1	INTRODUÇÃO . . . . .	24
3.2	MODELOS DE COMPUTAÇÃO EM NUVEM . . . . .	25
3.3	GERÊNCIA DE NUVENS COMPUTACIONAIS . . . . .	26
3.3.1	Provedores de IaaS . . . . .	27
3.4	GERÊNCIA DE RECURSOS EM NUVENS COMPUTACIONAIS . . . . .	28
3.4.1	Funcionalidades na Gerência de Recursos . . . . .	29
3.4.2	Objetivos na Gerência de Recursos . . . . .	30
3.5	GERÊNCIA DE RECURSOS BASEADA NA MIGRAÇÃO DE VMS . . . . .	31
3.6	CONSIDERAÇÕES FINAIS . . . . .	32
<b>4</b>	<b>ESTADO DA ARTE EM GERÊNCIA DE MIGRAÇÕES . . . . .</b>	<b>33</b>
4.1	INTRODUÇÃO . . . . .	33
4.2	GERÊNCIA DE MIGRAÇÕES INTRA-DC . . . . .	33
4.2.1	(Wood et al., 2007) . . . . .	34
4.2.2	(Xu e Fortes, 2011) . . . . .	36
4.2.3	(Beloglazov e Buyya, 2012) . . . . .	37
4.2.4	Outros trabalhos. . . . .	38
4.3	GERÊNCIA DE MIGRAÇÕES INTER-DC . . . . .	41
4.3.1	Reservoir . . . . .	41
4.3.2	InterCloud. . . . .	42
4.3.3	xVMotion . . . . .	44
4.3.4	ACTiCloud . . . . .	45

4.3.5	HeatSpreader . . . . .	48
4.4	DESAFIOS NA GERÊNCIA DA MIGRAÇÃO DE VMS . . . . .	49
4.5	CONSIDERAÇÕES FINAIS . . . . .	52
<b>5</b>	<b>MODELO PARA A GERÊNCIA DA MIGRAÇÃO DE MÁQUINAS VIRTUAIS . . . . .</b>	<b>53</b>
5.1	VISÃO GERAL DO MODELO . . . . .	53
5.2	IDENTIFICANDO A NECESSIDADE DE MIGRAÇÕES . . . . .	54
5.3	SELEÇÃO DE VMS A MIGRAR . . . . .	56
5.4	SELEÇÃO DE HOSTS DE DESTINO DE MIGRAÇÕES. . . . .	57
5.5	CONSIDERAÇÕES FINAIS . . . . .	60
<b>6</b>	<b>UMA ARQUITETURA PARA A GERÊNCIA DE MIGRAÇÕES . . . . .</b>	<b>61</b>
6.1	VISÃO GERAL DA ARQUITETURA. . . . .	61
6.2	DETALHAMENTO DOS COMPONENTES E SUAS INTERAÇÕES . . . . .	62
6.3	CONSIDERAÇÕES FINAIS . . . . .	65
<b>7</b>	<b>AVALIAÇÃO. . . . .</b>	<b>67</b>
7.1	INTRODUÇÃO. . . . .	67
7.2	MODELO DE SIMULAÇÃO. . . . .	67
7.2.1	Caracterização de VMs e Workloads . . . . .	68
7.3	MÉTODO DE AVALIAÇÃO . . . . .	69
7.4	EXPERIMENTO 1: TAXA DE SUCESSO EM MIGRAÇÕES . . . . .	70
7.5	EXPERIMENTO 2: QUANTIDADE DE HOSTS SOBRECARGADOS . . . . .	72
7.6	CONSIDERAÇÕES FINAIS . . . . .	74
<b>8</b>	<b>CONCLUSÕES . . . . .</b>	<b>76</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>78</b>

## 1 INTRODUÇÃO

Recursos como memória, largura de banda de rede e capacidade de processamento são essenciais para os serviços baseados em ambientes de computação em nuvem (Cloud Services - CS). Em geral, tais recursos são distribuídos na forma de Máquinas Virtuais (Virtual Machines - VMs). Para atender a uma quantidade diversa de consumidores, os CSs estão propensos a variações inesperadas na demanda de recursos. Como consequência, mecanismos que permitam a provisão dinâmica de recursos tornam-se necessários. A principal estratégia atualmente adotada é a migração de VMs, ou seja, a transferência de uma VM para outro *host* que consiga atender às demandas de recursos da VM.

A migração de VMs é uma tarefa que envolve, minimamente, duas decisões: qual VM está, atualmente, interferindo na provisão dos serviços e qual o destino adequado para reacomodar esta VM. Contudo, quanto mais CSs envolvidos, mais frequente é a necessidade de migrações e, consequentemente, maiores são os esforços necessários para se planejar migrações de VMs. Em um cenário que envolve tantos interesses distintos, gerenciar as operações de migração torna-se inevitável. Contudo, esse é um desafio ainda pouco considerado, estando limitado a esforços de coordenação de operações de baixo nível. Neste capítulo são apresentados: o contexto em que se insere a gerência das migrações de VMs, a descrição do problema, os objetivos que se procurou alcançar, as contribuições que advêm deste trabalho e a estruturação do documento desta tese.

### 1.1 CONTEXTUALIZAÇÃO

Plataformas para a Gerência de Nuvens (*Cloud Management Platforms* - CMP), via de regra, incorporam a migração de VMs no âmbito de uma funcionalidade designada gerência de recursos (Council, 2017). CMPs são o resultado da integração de diferentes ferramentas, dessa forma, as ações que contribuem para o planejamento e a execução da migração dependem de diferentes ferramentas e fontes de informação externas. Em decorrência, é natural que haja uma diversidade de estratégias para o planejamento e a execução de migrações (Freet et al., 2016).

A integração de diferentes (e heterogêneas) ferramentas aliada à necessária intervenção humana gera um cenário propenso a falhas, contribuindo para a produção de planos de migração falhos e de migrações cujas execuções trazem ainda mais prejuízos. Neste contexto, uma solução que habilite o controle e o monitoramento das migrações tem o potencial de diminuir (e até eliminar) os prejuízos decorrentes das migrações mal planejadas.

CMPs operam no gerenciamento de diferentes nuvens, podendo atender a distintos modelos. Pode-se considerar que uma CMP atenda a uma ou mais nuvens, e consequentemente a um *datacenter* (DC) ou mais. Neste contexto, tem-se três potenciais escopos para a migração de VMs: (a) intra-DC; (b) intra-nuvem e (c) inter-nuvem. No escopo (a), VMs migram apenas entre os *hosts* que compõem o DC. Em um cenário intra-nuvem, VMs podem migrar entre diferentes DCs, caso a nuvem seja composta de mais de um DC. No último escopo tem-se um cenário que é obrigatoriamente inter-DC. Este trabalho apresenta uma solução de gerenciamento de migrações que atende aos escopos identificados como intra-DC e intra-nuvem.

### 1.2 O PROBLEMA

Migrações de VMs demandam capacidade de processamento, armazenamento e largura de banda adicionais, sendo, portanto, operações que tendem a possuir alto custo computacional.

No entanto, são importantes soluções para lidar com o intrínseco dinamismo das cargas de trabalho que VMs podem assumir. De fato, as migrações de VMs têm sido a abordagem padrão para solução de problemas decorrentes da flutuação constante no uso dos recursos. Conforme descrito por 2, é possível identificar dois tipos de soluções para lidar com as variações constantes no uso de recursos, ou seja, da elasticidade de dados em Serviços Baseados em Nuvem (*Cloud Services* - CS): a otimização da operação de migração e a automação do planejamento de migrações.

No primeiro tipo de solução pode-se incluir algoritmos e heurísticas que visam diminuir os impactos na infraestrutura associada. Tem-se como exemplo o trabalho de (Clark et al., 2005) e o de (Mandal et al., 2016). Em geral, são trabalhos dedicados à redução de tráfego e do tempo dedicado à execução da migração.

Na outra categoria estão as estratégias para automação do planejamento de migrações. A principal ideia é que um plano bem elaborado impede que uma migração desnecessária seja executada. Conforme descrito na contextualização, as atuais soluções de planejamento de migrações têm se baseado em CMPs, situando a migração como uma tarefa complementar no gerenciamento de recursos. Em decorrência, o planejamento de uma migração torna-se dependente da utilização de diferentes ferramentas externas e da consequente intervenção de um operador humano.

Dois aspectos são relevantes para considerar que as atuais soluções de migrações são propensas a falhas: a fraca integração das ferramentas assim como das informações e a necessária intervenção humana no planejamento de migrações (Zheng et al., 2019; Wickboldt et al., 2014). Embora ferramentas como as CMPs sejam empregadas, é possível constatar que alguns desafios ainda não foram claramente solucionados. Conforme mencionado por (Zhang, 2018), usuários não são providos com mecanismos para o controle do processo de migração e também pouco tem sido feito para que uma migração incorpore, de fato, os requisitos dos consumidores. Ainda, (Boutaba et al., 2013) destaca os desafios associados à sobrecarga que uma migração causa, ressaltando que o uso eficaz das migrações é um desafio que demanda atenção. Conforme (Zheng et al., 2019; Liu et al., 2020), a ausência de controle podem induzir a planos de migração que produzam o indesejado efeito de *thrashing* que é a contínua migração de uma VM entre distintos *hosts*, influenciando na falta de recursos e também na geração de prejuízos ao ambiente.

A automação de planos de migração tem sido apresentada como uma abordagem para diminuir os impactos das migrações mal sucedidas (mal planejadas e, portanto, cuja execução gera ainda mais prejuízos). Entretanto, pouca atenção tem sido dada ao gerenciamento da migração, ou seja, ao controle e ao monitoramento da migração em diferentes etapas. De fato, a execução de uma migração demanda identificar: a necessidade de migrar, a escolha da VM que deve migrar e do destino para onde deve ser transferida. Diferentes migrações podem ocorrer simultaneamente, gerando impacto na disponibilidade dos recursos. Ou seja, a falta do gerenciamento estimula decisões de migração baseadas em informações equivocadas.

É necessário uma abordagem de automação do planejamento de migrações cujas decisões decorram de ações de gerenciamento, pois, do contrário, as migrações de VMs tenderão a gerar mais prejuízos na execução de CSs.

### 1.3 OBJETIVOS

O principal objetivo deste trabalho é a concepção de uma arquitetura de gerenciamento de migrações que possibilita a geração de planos de migração de VMs que sejam efetivos. Um plano de migração é efetivo quando o resultado da migração é satisfatório, ou seja, quando a instabilidade verificada no *host* que hospeda a VM é solucionada pela migração da VM. Do

contrário, se a migração produzir novas necessidades de migração, então, considera-se que a migração não foi efetiva. Contudo, a arquitetura é a base para solução de diferentes propósitos na migração de VMs: balizar eventos em *hosts* que podem requerer a execução de uma migração e localizar VMs migrantes, assim como *hosts* de destino que possibilitam a geração de um plano de migração efetivo. Este objetivo tem aspectos específicos que decorrem da concepção da arquitetura:

- ratificar a hipótese de que o gerenciamento de migrações de VMs produz sistemas mais estáveis, com a diminuição da necessidade de migrações;
- atestar que a integração de informações decorrentes da execução das migrações influencia em planos de migração efetivos;
- analisar e prover uma abordagem para a detecção de necessidades de migração;
- levantar o estado da arte e prover uma abordagem para a seleção de VMs candidatas a migrar;
- levantar o estado da arte e prover uma abordagem para a seleção de *hosts* de destino apropriados na construção de planos de migração.

#### 1.4 CONTRIBUIÇÕES

As principais contribuições dessa tese são:

- contextualização e sistematização de gerência de migrações como uma área funcional no âmbito do gerenciamento de recursos de uma nuvem;
- definição, simulação e avaliação de uma arquitetura que permita planejar e executar migrações de VMs, integrando diferentes ambientes destinados ao gerenciamento de nuvens.

#### 1.5 ORGANIZAÇÃO DO DOCUMENTO

Esta tese está organizada conforme a seguinte estruturação. No Capítulo 2 VMs são conceituadas assim como são apresentados os desafios para que a migração de VMs possa ser planejada e executada. No Capítulo 3 são apresentados conceitos fundamentais sobre virtualização e migração de VMs, sendo ressaltados o potencial e também os obstáculos para a adoção da migração de VMs como estratégia para a alocação dinâmica de recursos em nuvens. No Capítulo 4 são descritos trabalhos que contribuem para a adoção de uma abordagem para o gerenciamento da migração de VMs, destacando-se estratégias para a automação da migração de VMs e os mecanismos que contribuem na melhoria do desempenho na execução de migrações. No Capítulo 5 é apresentado um modelo para o gerenciamento da migração de VMs e no Capítulo 6 sua respectiva proposta de implementação, por meio de uma arquitetura. No Capítulo 7 é apresentado um modelo para os testes da arquitetura proposta e as respectivas métricas avaliadas. O Capítulo 8 apresenta as conclusões e perspectivas de continuidade e melhoria do trabalho.

## 2 CONCEITOS DE MÁQUINAS VIRTUAIS

Este Capítulo apresenta o conceito de Máquina Virtual, hipervisor e os desafios associados ao planejamento e à execução de migrações. Além disso, o custo para se planejar e executar migrações é discutido, assim, como as etapas que permitem a construção de planos de migração. Tais conceitos fundamentam a solução para o problema apresentado anteriormente. O que se pretende é apresentar uma solução para a Gerência da Migração de VMs, contudo, ressaltando que essa atividade ainda não existe de forma clara e que as atuais abordagens de solução apresentam falhas e limitações.

Este capítulo inicia com uma discussão sobre o que são Máquinas Virtuais e hipervisores, apresentados na Seção 2.1. A Seção 2.2 apresenta o que é migrar VMs e discorre sobre o seu potencial assim como os desafios que dificultam sua utilização. Na Seção 2.3 é discutido o custo computacional decorrente tanto do planejamento como da execução de uma migração. A Seção 2.4 ressalta as etapas que compõem o processo de planejamento de uma migração. Os desafios e as potencialidades da migração de uma VM são ressaltados nas considerações finais, na Seção 2.5.

### 2.1 MÁQUINAS VIRTUAIS E HIPERVISORES

De acordo com (Smith e Nair, 2005), uma VM é uma abstração de uma máquina real, tendo o potencial de mapear virtualmente os recursos associados a esta máquina. Três recursos permitem, invariavelmente, a concepção de uma VM: a memória, as conexões de rede e a capacidade de processamento. Cada um desses recursos está relacionado a um correspondente físico, ou seja, os recursos associados a uma VM são designados virtuais. A concepção de VMs decorre das ideias originadas do conceito de virtualização. Desde a década de 60, conforme mencionado por Cusumano (2010), já eram discutidas ideias para otimizar o uso dos recursos de um computador. O estabelecimento de um mapeamento entre recursos físicos e virtuais já era discutido. A virtualização, portanto, propõe um uso mais flexível dos recursos físicos, impulsionando sua utilização em um escopo amplo e abrangente, diminuindo a subutilização e até mesmo potenciais gargalos.

Em decorrência da estreita relação entre recursos físicos e virtuais, VMs são consideradas entidades convidadas de um *host*. Como cada *host* é o provedor dos recursos físicos para as VMs, são considerados os hospedeiros das VMs. Essa relação *hospedeiro-convidado* permite compreender o papel dinâmico de uma VM, ou seja, ela pode se tornar convidada em diferentes hospedeiros e, para tanto, é imprescindível que seu estado possa ser constantemente acessado.

Toda VM age como uma entidade independente, podendo inclusive operar sob um sistema operacional distinto de seu hospedeiro, tendo, portanto, um conjunto de instruções distintos para o acesso aos recursos físicos. Uma decorrência imediata desta característica é a capacidade que cada VM tem de operar de forma isolada, não interferindo e nem sendo interferida por outras VMs no mesmo hospedeiro. Essa propriedade das VMs é designada como isolamento.

Cada VM tende a ter um ambiente de execução de instruções próprio e que, invariavelmente é distinto do que é provido pelo hospedeiro. Em face dessa heterogeneidade, torna-se imperativa a existência de uma entidade que lida, de forma transparente com os distintos formatos de instruções. Nesse contexto, um elemento possui papel relevante, o *Hipervisor* ou o Monitor de VMs (Rosenblum e Garfinkel, 2005). Conforme descrito na Figura 2.1, trata-se de uma camada de virtualização que permite VMs, detentoras de distintas interfaces, de compreenderem instruções



particulares providas pelo ambiente em que estão hospedadas. Ou seja, o ambiente onde uma VM está sendo executada compreende as requisições que lhe são feitas e vice-versa, mesmo que os conjuntos de instruções de sistema sejam distintos. Esse mapeamento de instruções de uma plataforma para outra possui o suporte do *hipervisor*.

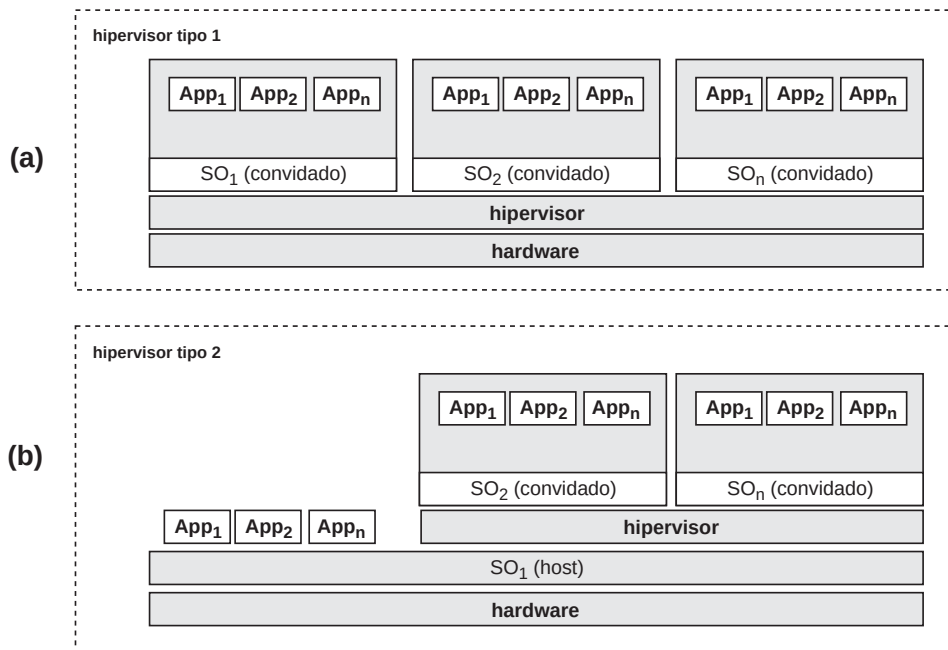


Figura 2.1: Interação de VMs mediada por um *hipervisor*.

*Hipervisores* podem provir de dois esquemas distintos: tanto podem ser oriundos de uma implementação sobre o *hardware* como também sobre um *software*. No primeiro caso, identificado na Figura 2.1(a) como *hipervisor* tipo 1, também designado como *bare-metal*, o conjunto de instruções do sistema convidado, ou seja, do ambiente virtualizado, é automaticamente interpretado para as instruções nativas do sistema. Essa interpretação ocorre diretamente ao nível de hardware, já que o próprio *hipervisor* torna-se elemento independente do Sistema Operacional (SO). Em contrapartida, como descrito na Figura 2.1(b), *hipervisores* do tipo 2 são um processo como qualquer outro do ambiente que hospeda as VMs. VMs baseadas em *hipervisores* do tipo 2 têm menor controle sobre o hardware subjacente, ao contrário dos *hipervisores* do tipo 1. No entanto, VMs que operam sob um *hipervisor* tipo 1 tendem a encontrar maior dificuldade para lidar com questões de compatibilidade de *hardware*.

Uma VM é um ambiente de suporte à execução de *software*, tendo o potencial de incrementar a utilização de recursos em diferentes níveis. Conforme descrito por Smith e Nair (2005), VMs podem operar sob duas perspectivas distintas. Na primeira, têm-se as VMs de processo que servem para possibilitar a execução de processos específicos em um Sistema Operacional. Em geral, visam suportar a execução de linguagens independentes de plataforma (por exemplo, Java, C#, Python, Perl, ...). Na outra perspectiva, têm-se as VMs de sistema que provêm a ilusão de um sistema completo, estimulando o compartilhamento efetivo de recursos físicos de um *host*. Uma VM de sistema contribui para a consolidação do conceito de isolamento e, portanto, da flexibilidade de operação simultânea de distintos ambientes de execução de tarefas.

A otimização do uso dos recursos tem, ao longo do tempo, permitido que VMs estejam associadas a distintos aspectos e interesses. VMs têm sido, por exemplo, exploradas no âmbito de:

- *Segurança Computacional*: possibilitando, através da propriedade de isolamento, prover caixas de areia (*sandbox*) que delimitam o escopo de ação em serviços sensíveis e que têm requisitos de controle de acesso rígidos;
- *Manutenção de Sistemas*: ainda tendo como base a propriedade de isolamento, VMs podem ser usadas para testes de novos recursos de um sistema sem que o sistema seja interrompido, permitindo que alterações sejam mais facilmente conduzidas;
- *Oferta Dinâmica de Recursos*: prover capacidade de computação sob demanda, na forma de VMs que podem ser criadas, destruídas ou redimensionadas conforme a necessidade de uso dos serviços por elas providos.

Imagem é uma representação estática da VM, ou seja, recursos até podem estar associados à VM, mas, sem que alguma referência ou valor estejam atribuídos aos recursos associados. Usualmente, a expressão *Virtual Appliance* é associada a uma imagem, sendo uma referência à capacidade de se tornar, mesmo que virtualmente, um instrumento para a concepção de serviços. Uma instância diz respeito a uma VM em execução, ou seja, a partir de uma imagem é possível atingir-se um estado em que uma VM já possua recursos alocados.

Ao longo do tempo de vida de uma VM, dois conceitos são relevantes para esta proposta. Um *snapshot* é um extrato de execução de uma VM, ou seja, um instante em que o estado de uma VM pode ser capturado. *Snapshots* representam um instante na execução das tarefas que foram delegadas a uma VM, ou seja, sua carga de trabalho (ou *workload*).

Na sua criação, uma VM precisa ser alocada a um *host*. A alocação é a operação que permite estabelecer um mapeamento entre os recursos físicos do *host* aos virtuais criados para a VM (Pietri e Sakellariou, 2016). Trata-se de uma operação associada a dois desafios. O primeiro diz respeito à definição de configurações apropriadas de VMs e o outro é relativo à determinação de quais *hosts* sejam adequados para acomodar tais VMs. Determinar uma configuração ideal para uma imagem de VM, de maneira que essa configuração seja potencialmente apropriada às demandas de recursos dos consumidores é o primeiro problema. Posteriormente, tem-se o outro desafio: como mapear de forma efetiva uma VM a um *host*? Esse desafio também é difícil já que interesses de diferentes atores precisam ser considerados, além de eventuais restrições que sejam relativas ao ambiente, há também questões legais que devem ser consideradas (Ponraj, 2019). Ainda, esse desafio diz respeito à capacidade de, ao longo da evolução de uso da VM, ser necessária uma realocação dessa VM.

Após serem alocadas, VMs podem receber cargas de trabalho. Contudo, é característico que o uso dos recursos varie de maneira imprevisível, tornando a construção de soluções para a falta ou mau uso de recursos um problema complexo. Neste contexto, quando a capacidade dos recursos de uma VM não puder ser mais atendida por um *host* tem-se, conforme descrito na Figura 2.2, duas opções: o redimensionamento ou a migração.

Um redimensionamento é um ajuste do mapeamento de recursos que não exige explicitamente o deslocamento da associação VM-*host* (Aldossary e Djemame, 2018). Contudo, o redimensionamento de uma VM pode requerer que outras VMs migrem, caso a capacidade contratada de outras VMs não possa continuar sendo atendida. Uma migração é a transferência de uma VM para outro *host*, requerendo que um novo mapeamento dos recursos virtuais da VM seja produzido, agora correspondendo aos recursos físicos do novo *host*.

## 2.2 A MIGRAÇÃO DE VMS

Conforme ressaltado por Boutaba et al. (2013), uma migração é a movimentação de uma unidade que possui um sistema operacional com execução em curso e, por conseguinte,



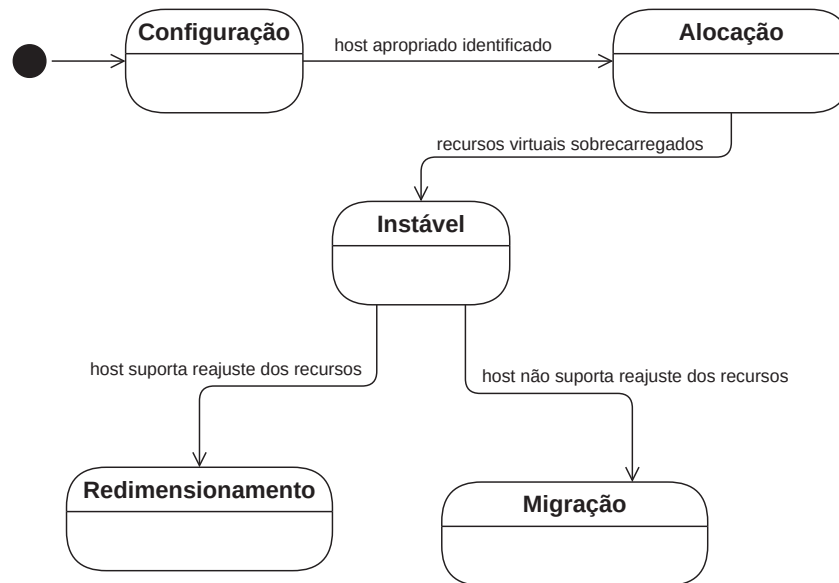


Figura 2.2: Potenciais estados de uma VM.

munido de inúmeros processos. Todos esses processos representam tarefas, em sua maioria associadas a interesses de usuários e que estão ligados a recursos (físicos e virtuais). Portanto, reforçando o conceito de migração de Medina e García (2014), migrar uma VM é, minimamente, uma operação de três etapas:

- capturar (ou serializar) o estado dos recursos na VM que será migrada. Nesse momento, não só o estado dos recursos, mas, todo o ambiente de execução, portanto, o sistema operacional da VM precisa ser capturado (Boutaba et al., 2013);
- transferir essa VM previamente capturada para um *host* de destino que seja compatível com as restrições técnicas e estratégicas da VM que está sendo migrada;
- recuperar o estado da VM e retomar sua execução no novo ambiente. Neste momento, os recursos físicos do novo *host* precisam ser mapeados aos recursos virtuais da VM e, especialmente, o *hipervisor* do destino precisa retomar o controle desta VM.

De fato, o *hipervisor* possui papel central na condução da operação de migração de VMs. Esta constatação decorre do fato de que um *hipervisor* é responsável por controlar diretamente o estado de três recursos virtuais de uma VM: a memória, as conexões de rede e a correspondente capacidade de processamento alocada. Ou seja, conforme descrito na Figura 2.3, migrar uma VM de um *host* para outro é eminentemente uma tarefa conduzida por *hipervisores*, cujo objetivo é capturar o estado de tais recursos e transferi-los para um destino.

Há, basicamente, duas abordagens para a Migração de VMs: Migração a Quente (*Hot/Live Migration*) e Migração a Frio (*Cold/Non-Live Migration*) (Svard et al., 2014). Quando a carga de trabalho de uma VM mantém-se em execução enquanto ela é migrada, tem-se uma Migração a Quente. Ou seja, seu propósito é manter o estado dos serviços praticamente estáveis, provendo transparência de localização aos consumidores. Por outro lado, na Migração a Frio a VM é desligada, sendo transferidas todas as informações necessárias para que ela possa reiniciar suas atividades no novo destino. Como consequência, o consumidor pode experimentar algumas interrupções na oferta do serviço. Apesar de propiciar maior simplicidade no processo de transferência, a Migração a Frio tem sido pouco adotada, e isso se deve ao fato de que a VM que está sendo transferida possui grande probabilidade de manter-se inoperante por um longo

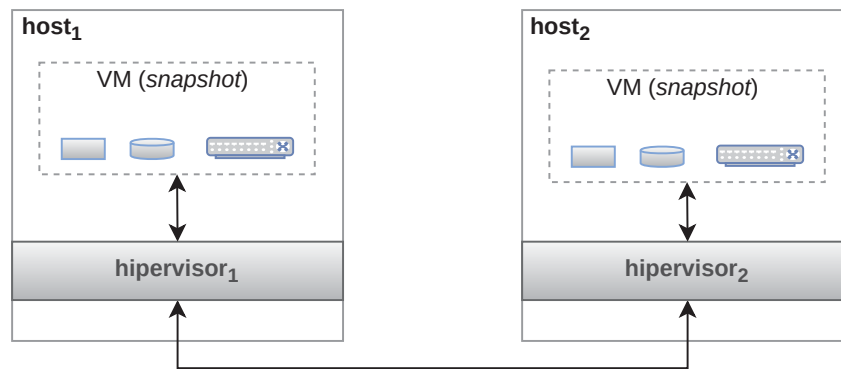


Figura 2.3: Migração de uma VM mediada por hipervisores.

período de tempo. Essa característica, via de regra, é indesejada em cenários de provisão de serviços (Medina e García, 2014), sendo, portanto, ao longo deste trabalho, consideradas apenas propostas de Migração a Quente de VMs.

Durante a migração de uma VM, seu estado completo precisa ser transferido. Este estado inclui informações dos recursos que são usados pela carga de trabalho da VM, representando informações que incluem: a capacidade de processamento alocada para a VM, a memória usada e o estado das conexões de rede. Dois formatos de armazenamento influem nas informações de memória: o armazenamento permanente e o volátil. Em geral, *Datacenters* (DCs) operam com modelos compartilhados de armazenamento permanente, portanto, nesses casos a transferência desse conteúdo é mais simples. Entretanto, a transferência da memória RAM requer maior cuidado. As tarefas da VM continuam em execução, forçando, portanto, que páginas de memória sejam modificadas. Em decorrência destas características, a transferência da memória é um fator crucial na migração de uma VM.

Para se efetuar uma Migração a Quente de uma VM, três fases de transferência de memória podem ser identificadas (Svard et al., 2014),(Kapil et al., 2013):

- **push**: trata da preparação do *snapshot*, possibilitando que o estado atual da VM seja capturado e efetuando a cópia de certas páginas de memória para o destino.
- **pull**: a VM inicia rapidamente sua operação no destino, assim que a VM na origem é suspensão. Durante a suspensão, um conteúdo mínimo da VM é capturado (designada por Ahmad et al. (2015) como estado mínimo) e transferido para o destino.
- **stop-and-copy**: além da preparação e da efetivação da transferência, a migração de uma VM exige que em algum instante a VM de origem seja definitivamente interrompida. No tocante à transferência de memória, essa interrupção ocorre ou quando todas as páginas de memória tiverem sido copiadas ou quando um determinado limiar (*threshold*) de cópias tiver sido atingido. A essa fase de interrupção tem-se designado o termo *stop-and-copy*.

A combinação de duas dessas fases estabelece dois diferentes formatos de Migração a Quente (Svard et al., 2014). O primeiro é designado como *Pre-Copy*, uma combinação das fases *push* e *stop-and-copy*. Esta estratégia foi concebida por Clark et al. (2005), sendo um tipo de migração em que a transferência de memória determina o início da migração. Neste caso, ocorre inicialmente uma cópia iterativa de páginas de memória até que algum critério seja encontrado. A VM na origem não é interrompida, ou seja, outros recursos tais como conexões de rede e o próprio sistema de arquivos mantêm-se em uso. Considera-se que *Pre-Copy* tenha um tempo

de migração maior quando as ações na VM de origem são eminentemente de escrita (ou seja com maior alteração de conteúdo das páginas), no entanto, é a estratégia com a capacidade de fornecer a um consumidor a percepção de não interrupção da oferta dos serviços baseados na VM (Sharma e Chawla, 2013). Ao final, quando a VM de origem precisar ser definitivamente interrompida, ocorre a fase *Stop-and-Copy* em que as páginas de memória que foram modificadas ao longo do processo (ou seja, as páginas "suja") são definitivamente transferidas para o novo destino. Neste momento, o *hipervisor* no destino assume a VM e retoma suas atividades a partir do estado que foi transferido.

O segundo tipo de Migração a Quente é designado como *Post-Copy*, uma combinação das fases *pull* e *stop-and-copy*. De imediato, a VM é suspensa na origem e um estado mínimo é capturado. Esse estado mínimo é transferido para o destino e a VM é reiniciada no destino. Para reiniciar no destino, todas as referências à VM são direcionadas para o novo endereço que é gerenciado pelo *hipervisor* no destino. Conforme Hines e Gopalan (2009), o propósito nessa estratégia é adiar ao máximo a transferência de memória, requerendo o início da transferência tão logo ocorra uma falta de página. Neste instante, a fase *Stop-and-Copy* inicia, prevendo cópias iterativas até que todas páginas tenham sido copiadas. Uma consequência inevitável dessa abordagem é que a recuperação de páginas na origem força, em algum momento, a interrupção da provisão dos serviços baseados na VM. Entretanto, conforme citado por Sharma e Chawla (2013), nesta estratégia de migração, as páginas de memória são copiadas apenas uma vez (à medida que faltas de página ocorrem no destino).

Conforme descrito na Figura 2.2, a migração de uma VM decorre da incapacidade de um *host* em atender à possibilidade de estender a capacidade dos recursos contratados na VM. Essa decisão pode advir de duas situações: restrições contratuais e limitações no ambiente. No primeiro caso, o consumidor que contratou a VM pode ficar impossibilitado de requisitar a extensão dos recursos em decorrência de limitações tanto financeiras como técnicas. A segunda restrição ocorre quando outras VMs no mesmo *host* não podem, contratualmente, abdicar do que foi contratado.

A utilização de VMs tornou-se popular no contexto de *datacenters* (DCs) e *clusters*. A migração de instâncias de sistemas entre distintos *hosts* é uma ferramenta útil para administradores de DCs. Ela permite uma clara separação entre *hardware* e *software* e facilita, conforme descrito por Sapuntzakis et al. (2002): o compartilhamento de recursos, o gerenciamento de faltas, o balanceamento de carga, o gerenciamento da eficiência energética, o controle da dissipação de calor e a manutenção de sistemas.

Apesar das contribuições, as migrações de VMs podem contribuir para a ocorrência de instabilidades no sistema. Conforme ressaltado por Zhang et al. (2018b), as migrações afetam todos os envolvidos na operação: a VM que migra, o *host* de origem, o *host* de destino e as VMs co-localizadas em ambos os *hosts*. De fato, há um inevitável incremento no consumo dos recursos, ocasionando, situações indesejadas tais como: a interrupção momentânea de CSs associados às outras VMs no ambiente (*downtime*), a queda de desempenho (*slowdown*) e o aumento no consumo de energia e da dissipação de calor (Zhang et al., 2017; Ayoub et al., 2019; Giles et al., 2020).

Para (Zhang et al., 2018b; Ayoub et al., 2019), as interferências que são causadas, em decorrência das migrações, podem ser reduzidas a três tipos de sobrecargas: de processamento, de rede e de espaço. No primeiro caso, a sobrecarga ocorre por três razões: (a) o *hipervisor*, que atua como um processo *daemon* já determina uma sobrecarga natural no consumo de recursos; (b) a migração determina uma natural duplicação da execução do mesmo ambiente da VM, especialmente quando se trata de migração a quente, em que o mesmo ambiente da VM pode

manter-se em execução em dois *hosts* por algum período de tempo e (c) em casos de estratégias tais como a deduplicação e a compressão de dados.

No segundo caso, tem-se a natural sobrecarga da rede. Consoante o comportamento dos *workloads* em execução nas VMs, têm-se fluxos de dados transferidos entre diferentes *hosts*. Essa situação torna-se ainda mais complexa em casos de alta geração de páginas sujas de memória. E, no terceiro caso, é natural que algum espaço extra seja necessário para contemplar a transferência de uma VM de seu *host* de origem para o destino.

Para diminuir o impacto que uma migração pode causar, diferentes métricas podem ser utilizadas para se prever o custo que uma migração pode ter. Essa é a estratégia adotada em abordagens que buscam otimizar as migrações através do estabelecimento de um Problema de Otimização Multi-Objetivo (*Multi-Objective Optimization Problem* - MOP). No entanto, também podem ser adotadas na construção de políticas para o planejamento de migrações. Na próxima subseção são apresentadas diferentes métricas que podem ser utilizadas no suporte ao planejamento de migrações.

### 2.3 CUSTOS DE UMA MIGRAÇÃO

As diferenças de desempenho das estratégias de migração a quente estimulam que discussões sobre as métricas para avaliação destas estratégias sejam feitas. De fato, muitos autores propuseram extensões a cada estratégia sendo, em geral, orientadas pela otimização de algum aspecto ou cenário particular. Por exemplo, Hines e Gopalan (2009) propuseram um esquema de *pre-paging* visando diminuir a quantidade de faltas de página em uma estratégia *Post-Copy* de migração.

Esforços como o de Hines e Gopalan (2009) reforçam uma discussão que tem se tornado cada vez mais frequente no tocante à migração de VMs: que métricas são, essencialmente, norteadoras da avaliação das estratégias de migração. Strunk (2012) apresentam em seu trabalho uma taxonomia com os tipos de custos que influenciam na migração de VMs e que têm sido investigados em diferentes trabalhos. Para os autores existem três categorias de custos: custos relativos ao desempenho da migração, custos relativos à perda de desempenho da VM durante a migração e custos associados à sobrecarga de consumo energético.

Adicionalmente, em outro trabalho, Zhang et al. (2018b) identificam quatro métricas que são usadas na avaliação do custo de uma migração: tempo total da migração, *downtime*, total de tráfego usado na rede e a degradação do sistema (*slowdown*). Conforme mencionado por Ayoub et al. (2019), algumas métricas também podem ser consideradas como um efeito que deve ser impedido. Por exemplo, o *downtime* ou a interrupção momentânea de um CS, em decorrência de alguma instabilidade no *host* é uma métrica que precisa sempre ser minimizada. Ou seja, boas estratégias de migração seriam aquelas que diminuem a ocorrência de *downtime*, assim como de todas as outras métricas mencionadas por Zhang et al. (2018b).

As métricas sugeridas por (Zhang et al., 2018b) também são consideradas por (Strunk, 2012). Na Figura 2.4 são apresentadas as métricas de ambos os autores, juntamente com a perspectiva de sobrecarga estabelecida por (Ayoub et al., 2019). Observa-se que há dois tipos de perdas que têm sido frequentemente avaliadas durante a migração de uma VM: desempenho e sobrecargas. O desempenho de uma migração é um dos aspectos mais investigados. Um parâmetro, dentro desse contexto, tem sido recorrentemente mencionado. É o tempo total de migração, ou seja, quanto tempo é gasto para que uma VM seja efetivamente interrompida na origem e tenha sua execução reiniciada no destino.

No tocante à queda de desempenho de uma VM durante o processo de migração, os autores mencionam que a influência principal decorre da quantidade de páginas sujas que

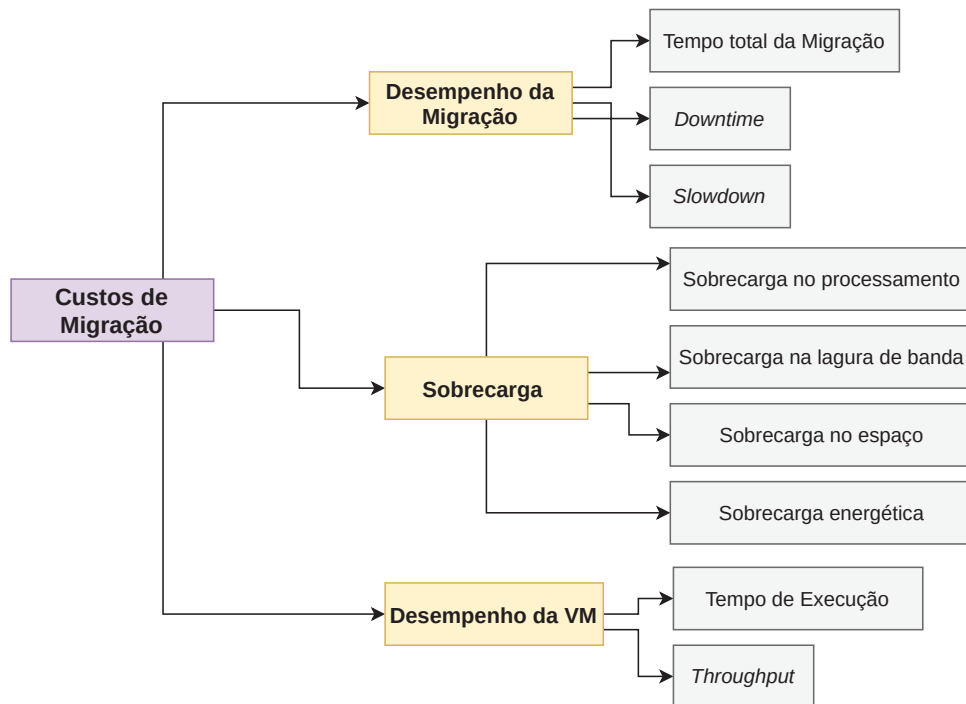


Figura 2.4: Custos de Migração conforme (Strunk, 2012; Zhang et al., 2018b; Ayoub et al., 2019).

precisam ser tratadas. Quanto maior a ocorrência e a necessidade de tratá-las, maiores são as perdas que podem ser observadas em dois parâmetros:

- **Tempo de execução:** em quanto tempo a execução da migração ocorre. Esse tempo pode ser influenciado pela necessidade de tratar páginas sujas.
- **Throughput:** o quanto os processos que fazem parte do *workload* da VM fluem, ou seja, se conseguem alcançar o seu intento.

O consumo energético tem sido tratado por diferentes autores. Embora (Strunk, 2012) não mencione explicitamente em sua proposta de taxonomia, os esforços associados à otimização do desempenho da migração têm sido cruciais na diminuição do consumo energético. Os trabalhos de (Baccarelli et al., 2015; Ayoub et al., 2019) reforçam essa constatação. A proposta dos autores é a de diminuir o consumo de energia pela minimização do tempo de migração. Neste caso, uma estratégia para diminuir a quantidade de páginas sujas foi concebida e, como consequência, os autores detectaram uma diminuição do uso de largura de banda. O resultado foi a minimização do consumo de energia durante uma migração de VM.

A taxonomia de custos proposta por Strunk (2012) também ressalta quais categorias de parâmetros influenciam os custos de migração. Há, conforme os autores três categorias:

- **Parâmetros associados ao *host*:** recursos de cada *host* (na origem ou destino) influenciam na migração. Por exemplo, poucos ciclos de CPU disponíveis implicam em um tempo total de migração maior.
- **Parâmetros relacionados à VM:** características específicas de uma VM interferem diretamente no tempo total de migração assim como no *downtime*. Por exemplo, quanto menor for o espaço de armazenamento volátil destinado a uma VM, maior a dificuldade de um *hypervisor* de atender a requisições de páginas de memória e, por conseguinte, maiores são as chances de que páginas na origem precisem ser atualizadas antes de

serem transferidas. A consequência é que, no destino, a quantidade de páginas sujas tende a crescer.

- **Parâmetros associados à infraestrutura:** um exemplo da influência dos parâmetros de infraestrutura é a largura de banda disponível. Quanto maior, mais rápida é a migração.

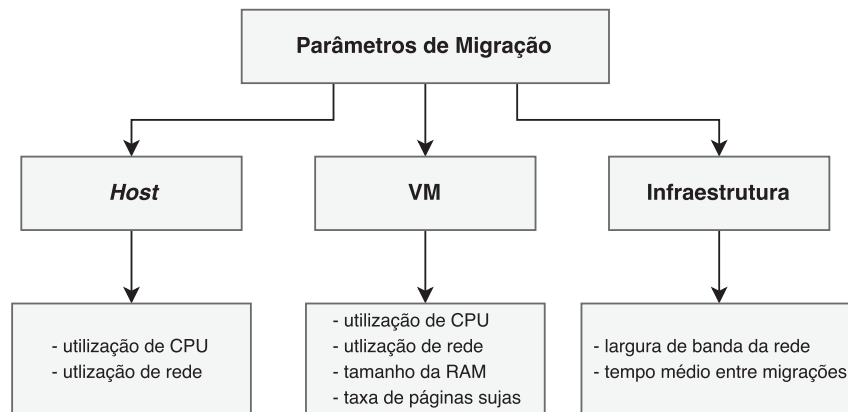


Figura 2.5: Parâmetros que influenciam na determinação dos custos de migração.

Cada estratégia distinta de migração pode apresentar vantagens em decorrência dos cenários. Por exemplo, a migração *Post-Copy* tende a impor maior *downtime*, em decorrência de eminente necessidade de tratar páginas sujas. Contudo, essa afirmação decorre de um cenário em que a quantidade de operações de escrita sejam superiores às de leitura no espaço de memória da VM de origem. Outro aspecto que merece ser investigado é o potencial de cada estratégia. Conforme (Ahmad et al., 2015), na migração *Pre-Copy* é possível obter maior confiabilidade contra falhas de sistema, já que sempre existirão duas cópias de páginas de memória em ambos os lados.

Reconhecer a existência de custos na migração de VMs é importante para lidar com o gerenciamento das atividades associadas à migração. Contudo, apesar da importante atuação dos *hipervisores* no controle e gerenciamento dos recursos no âmbito de suas VMs locais, é possível depreender que é importante que uma coordenação para a migração de VMs seja claramente estabelecida. Essa tarefa tem, de acordo com Smith e Nair (2005), dois cenários potenciais: um em que VMs são transferidas no âmbito de uma mesma rede local e outro em que domínios administrativos distintos estão envolvidos. Em cada caso, a transferência dos recursos virtuais de uma VM tende a encontrar obstáculos, decorrentes da complexidade em capturar o estado dos recursos virtuais.

## 2.4 CICLO DE VIDA DE UMA MIGRAÇÃO

Conforme Mishra et al. (2012); Medina e García (2014), para que uma migração de VMs seja planejada, é necessário que três passos sejam cumpridos: detectar e analisar um evento que demanda a ocorrência da migração; selecionar uma VM migrante e escolher um novo *host* para acomodar a VM que será migrada. Tais passos são elementares para a produção de um plano de migração, ou seja, um roteiro que retrate uma boa solução para a necessidade de migração inicialmente detectada, mas, que não incorra em prejuízos para o destino que receberá a VM.

O primeiro passo do ciclo de vida de uma migração de VMs envolve duas ações: monitoramento das condições de operação de cada *host* e a constatação da necessidade de migração de uma VM. Para que a primeira ação ocorra, é necessário que dados sobre a



demanda dos recursos de cada VM sejam coletados periodicamente. Em conjunto, indicadores de desempenho podem ser produzidos, possibilitando que se constate a necessidade de uma migração.

Em geral, a sobrecarga de algum recurso em um *host* é sinal de alerta e serve como gatilho para que se decida se uma migração deve ocorrer. Contudo, para (Pietri e Sakellariou, 2016), quatro tipos de gatilhos podem exigir uma migração: baseados em algum evento, periódicos, híbridos e baseados em *thresholds*. No primeiro caso, eventos como o encerramento ou a criação de uma VM podem exigir remapeamentos dos recursos. Pode-se também considerar que remapeamentos sejam periódicos, por exemplo, com o objetivo de manter sempre os *hosts* de um ambiente de nuvem dentro de um determinado patamar de consumo de energia, por exemplo, para manter o balanceamento ou a consolidação dos *hosts*. Ou também é possível combinar eventos específicos, tais como o encerramento de uma VM a outras checagens, tais como a verificação do atendimento aos objetivos descritos em SLAs. E, ainda, o gatilho mais comum é a averiguação periódica de limites de segurança pré-definidos (*thresholds*) e, em caso de extrapolação desses limites, planejar a execução de uma migração.

Três abordagens podem ser adotadas para a coleta das demandas: pró-ativa, reativa ou programada. Técnicas para a predição do uso dos recursos podem ser utilizadas, permitindo que uma análise da tendência do comportamento dos recursos antecipe a identificação de desequilíbrios na oferta de recursos. Em particular, ações pró-ativas contribuem para decisões consistentes de migração (Gmach et al., 2009). Ou seja, baseadas na confirmação de que o padrão de comportamento do ambiente decorre de um desequilíbrio que precisa ser corrigido e não de um desvio momentâneo (ou falso alarme), causado por um pico de uso dos recursos (Wood et al., 2007; Goumas et al., 2017). No entanto, a detecção pró-ativa requer maiores esforços, tais como: a coleta constante de dados, usualmente em intervalos designados janelas de tempo (Wood et al., 2007; Beloglazov e Buyya, 2012) e também o processamento desses dados em intervalos periódicos. A detecção reativa não é baseada em análises estatísticas, gerando, portanto, resultados mais rápidos sobre o impacto de cada gatilho de migração. No entanto, não consideram que um gatilho pode ser caracterizado como um falso alarme.

O resultado da análise dos gatilhos também permite averiguar o impacto que a migração pode causar, dando a possibilidade de que se atenda apenas aos planos de migração que não redundem em prejuízos ao ambiente. De fato, é consensual entre diferentes autores (Alicherry e Lakshman, 2012; Beloglazov e Buyya, 2013; Ayoub et al., 2019) que migrações precisam ser cuidadosamente consideradas e que a frequência de sua ocorrência deve considerar questões como: o custo da operação e o impacto no desempenho da execução de outras aplicações.

O planejamento de uma migração tem continuidade com a determinação de que VM precisa migrar. Como este passo decorre da detecção de problemas em um *host* (geralmente, uma sobrecarga), deve-se iniciar pela inspeção de quais VMs estimulam a ocorrência do problema. Conforme Wood et al. (2007), diferentes estratégias podem ser usadas para a escolha de uma VM migrante e que, nem sempre, a abordagem mais óbvia, como, por exemplo, escolher a VM que mais consome recursos de processamento, é a melhor solução (Basu et al., 2019).

Há outros aspectos que também precisam ser considerados: se o cenário envolve uma VM com alto consumo de recursos e se a VM possui relações de afinidade com outras VMs do ambiente. Em ambos os casos, a escolha de uma VM com muito consumo de recurso (por exemplo, memória) ou que tem um grupo de VMs associadas pode impactar tanto no uso da largura de banda da rede como de CPU e se tornar, portanto, desvantajoso.

A escolha de um destino adequado para a VM migrante conclui a elaboração do plano de migração. Dois aspectos influenciam nesta escolha: a capacidade disponível de recursos e as questões associadas à afinidade. As alternativas mais simples para a escolha de um destino

baseiam-se no critério "o primeiro que atende é o primeiro que é escolhido" (*First Come First Served* - FCFS), sendo usual que o destino de uma migração seja exatamente o *host* com a maior capacidade disponível de recursos. No entanto, apenas considerar a disponibilidade dos recursos no destino é um critério limitado (Mishra et al., 2012; Beloglazov e Buyya, 2013; Basu et al., 2019). Apesar de ser o destino com mais recursos disponíveis, é possível que a nova VM afete o desempenho de outras VMs, tornando o *host* rapidamente sobrecarregado.

Outro critério usualmente considerado é a afinidade. Nesse caso, é possível considerar a possibilidade e a necessidade de co-localização (*co-location*) da VM migrante com uma ou mais VMs no destino. É possível que restrições, especialmente no âmbito legal, impeçam que duas VMs coexistam no mesmo ambiente. Por exemplo, em casos de quebra de relações, uma VM contratada por um consumidor iraniano poderia ter restrições em coexistir com uma VM contratada por um consumidor norte-americano. Por outro lado, é possível que haja VMs que precisem coexistir com outras, requerendo que as suas parceiras façam parte do mesmo plano de migração. Esse é o caso de serviços que foram decompostos em diferentes unidades e que necessitam um do outro para atender às necessidades dos consumidores.

Considera-se que um plano de migração seja eficiente quando os objetivos de gerenciamento<sup>1</sup> são atendidos tanto na elaboração como na execução dos planos de migração. Para lidar com tais objetivos, cada etapa do ciclo de vida de uma migração emprega estratégias distintas.

Os objetivos de gerenciamento também são utilizados na identificação de VMs migrantes. No entanto, o custo de uma migração tem sido considerado um objetivo relevante para essa etapa. O impacto que uma migração pode causar na execução de outras tarefas e no uso dos recursos do ambiente, tais como largura de banda, capacidade de processamento e memória devem ser levados em consideração, permitindo antecipar o quanto a migração de determinada VM pode efetivamente satisfazer a um ou mais objetivos e ao mesmo tempo como essa decisão influencia no ambiente. Beloglazov e Buyya (2012) apresentam um modelo dos custos de uma migração, considerando que a redução do consumo de energia é um objetivo de gerenciamento. Além desse aspecto, o modelo também incorpora os custos com as violações de SLA e dessa forma, torna possível prever se uma migração incorrerá em custos excessivos.

Em outro trabalho, Dargie (2014) ressalta que a ordem das migrações influencia no custo da migração, sendo, importante classificar os planos de migração conforme seu custo ou de acordo com o impacto em algum objetivo específico. Para Zhao et al. (2019), o ponto de vista do consumidor tem sido negligenciado nos atuais modelos para estimação do custo de uma migração, sugerindo que estimar o desempenho da VM migrante também deve ser considerado. Ou seja, o desempenho da VM após a migração precisa ser mantido, pois, dessa forma o consumidor continuará a ter a mesma qualidade de experiência que já estava recebendo.

## 2.5 CONSIDERAÇÕES FINAIS

VMs possibilitam que recursos físicos de um *host* que poderiam ser subutilizados, não sejam desperdiçados. VMs são, hoje, uma tecnologia popular, adotada em diferentes contextos com o propósito de lidar com recursos computacionais. Elas têm sido a tecnologia chave para nuvens, possibilitando que os recursos computacionais em um DC sejam melhor utilizados. Em decorrência, é possível constatar sua relevância e, conseqüentemente, a importância de se prever estratégias que contribuam para que seus propósitos sejam, de fato, alcançados.

---

<sup>1</sup>Pietri e Sakellariou (2016) usam a expressão objetivo de otimização, uma alusão ao fato de que os objetivos são úteis para se alcançar resultados ótimos.



VMs representam virtualmente recursos físicos de um *host*, possibilitando que mapeamentos sejam feitos de maneira aleatória e, portanto, que diferentes tipos de máquinas possam ser concebidas.

Ao longo do tempo, recursos virtuais podem atingir limites considerados críticos. Esse fato exige que a VM seja transferida para outro *host*. De maneira simplificada, essa tarefa é executada por *hipervisores*, ou a camada de *software* que gerencia as VMs.

A migração de uma VM pode ocorrer de duas formas: com a interrupção das tarefas que estavam associadas à VM ou sem a interrupção. No primeiro caso (também chamado de migração a frio), as VMs são totalmente transferidas para o destino e as suas tarefas só são retomadas após o término da migração. No segundo caso (também chamado de migração a quente), as tarefas continuam sendo executadas apesar da migração. É possível não interromper as atividades caso consideremos que no início do processo de migração uma pequena, mas, suficiente parte da memória e do processamento da VM seja transferido.

Migrações a frio pode ser consideradas mais rápidas, pois não se considera a necessidade de lidar com páginas "sujas", decorrentes de processamento executado após o início de uma migração. No entanto, têm como preço o fato de interromper completamente o serviço que estava em execução na VM.

Migrações estão associadas a um ciclo de vida baseado em três passos: quando migrar, quem migrar e para onde. Tais passos possibilitam a construção de um plano de migração.

### 3 NUVENS COMPUTACIONAIS

Este capítulo apresenta as nuvens computacionais, destacando o papel das VMs como elemento essencial para que serviços possam ser distribuídos a distintos consumidores. Esta discussão no capítulo está organizada nas seguintes seções. A Seção 3.1 traz um conceito e as propriedades relativas a uma nuvem computacional. A Seção 3.2 descreve as diferentes possibilidades de se organizar um ambiente de computação em nuvem. A Seção 3.3 destaca estratégias que são adotadas para a gerência de ambientes de computação em nuvem. Na Seção 3.4 são ressaltados os desafios da gerência de recursos em ambientes de computação em nuvem. A Seção 3.5 destaca a relevância das migrações de VMs como estratégia para a gerência dinâmica de recursos. E, ao final, as considerações finais do capítulo estão na Seção 3.6.

#### 3.1 INTRODUÇÃO

O *National Institute of Standards and Technology* (NIST) (Mell e Grance, 2011) identifica a Computação em Nuvem como um modelo que facilita o desenvolvimento e a distribuição de serviços, por meio do acesso ubíquo e sob demanda a recursos. De fato, tal modelo contribui para que serviços de Internet não encontrem obstáculos tais como alto investimento de *hardware* dedicado e também em pessoal qualificado. Além disso, não requer que recursos sejam super-dimensionados a fim de evitar interrupções.

Nuvens são compostas de *datacenters* (DCs). DCs são ambientes projetados para agrupar servidores, dotando-os de infraestrutura de comunicação e armazenamento. Cada servidor (ou *host*) possui recursos de *hardware*, tais como armazenamento, capacidade de processamento e para execução de tarefas de rede que, tipicamente, têm o potencial de atender às necessidades de diferentes usuários em diversas localizações.

A virtualização é a tecnologia que permite o uso eficiente dos recursos de *hardware*, possibilitando mapeamentos dinâmicos de recursos físicos em correspondentes virtuais, ou seja, clientes e serviços de uma nuvem usam dos recursos por meio de VMs. Serviços baseados em Nuvem (*Cloud Services - CS*) são, portanto, aplicações desenvolvidas para tirar o maior proveito dessa capacidade de se obter, dinamicamente, recursos.

Talvez a principal contribuição da Computação em Nuvem seja a flexibilização da comercialização de recursos e serviços. Ambos podem ser adquiridos em um formato *pay-as-you-go*, ou seja, o consumidor adquire apenas o que precisa para o momento em que se faz necessário. Entretanto, outras características podem ser destacadas, conforme mencionado por Mell e Grance (2011): *multi-tenancy* e elasticidade.

Dá-se o nome de *multi-tenancy* à necessidade de lidar com diferentes consumidores para o mesmo conjunto de recursos (Takahashi et al., 2012). Cada consumidor de CSs é, tipicamente, designado como inquilino (*tenant*) de uma nuvem computacional, sendo necessária a divisão dos recursos físicos requisitados por cada um, em correspondentes virtuais. A elasticidade diz respeito ao acesso rápido aos recursos da nuvem, contudo com a liberdade de desfazer essa associação a qualquer instante (Galante e de Bona, 2012; Herbst et al., 2013). Essa característica possibilita que uma nuvem computacional seja flexível o suficiente para que recursos sejam dimensionados e alocados à medida que um consumidor necessite.

### 3.2 MODELOS DE COMPUTAÇÃO EM NUVEM

A flexibilidade e o dinamismo na oferta de recursos dá aos provedores possibilidades distintas de desenvolvimento de CSs. A abrangência de tais possibilidades é compreendida com o estabelecimento dos modelos em que CSs podem ser desenvolvidos em uma nuvem. Neste trabalho, os conceitos introduzidos pelo NIST em (Group et al., 2013) são utilizados para descrever: entidades que interagem no desenvolvimento, provisão e manutenção de CSs; modelos de CSs que, minimamente, podem ser oferecidos aos consumidores e o modelos sob os quais as nuvens podem ser implementadas.

Três entidades envolvidas na provisão de serviços baseados em nuvens (CSs) e com papéis distintos, são identificadas neste trabalho: *Cloud Consumer* (Cc), *Cloud Provider* (CP) e *Cloud Service Provider* (CSP). Ccs são os consumidores de um CS, podendo ser um consumidor humano ou até mesmo um CSP, quando um CS necessita consumir outro CS. Para efeitos de simplificação, ao longo deste trabalho, será designado apenas como consumidor. CP é a entidade responsável por gerenciar e controlar a infraestrutura associada à nuvem computacional. Ou seja, o responsável pelos *hosts* do DC de onde provêm os recursos físicos da nuvem assim como da integração de diferentes DCs, caso a nuvem seja composta por mais de uma nuvem. E o CSP é a entidade responsável por construir, a partir do mapeamento dos recursos físicos ofertados por CPs, serviços que são providos aos consumidores.

Recursos físicos de uma nuvem são, usualmente, abstraídos em VMs e comercializados por CPs. Tipicamente, tais VMs dão liberdade para que CSPs possam desenvolver diferentes CSs. Conforme o NIST (Group et al., 2013), deve-se, minimamente, considerar três tipos de serviços, ou modelos de serviços: *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Software as a Service* (SaaS). No caso de IaaS, recursos de *hardware* são abstraídos e oferecidos como VMs aos consumidores. Tais VMs usam da infraestrutura do CP, executando eventuais cargas de trabalho. Para o modelo PaaS, CPs disponibilizam ambientes em que consumidores possam implementar serviços e disponibilizá-los na nuvem. Serviços distribuídos sob o modelo SaaS podem ser acessados pelos consumidores, contudo, sem que haja permissão para controlar como esse serviço é executado, ou seja, o consumidor não escolhe a VM e nem o *host* onde o serviço é executado.

CSPs podem desenvolver e distribuir SaaS tanto através de IaaS como de PaaS. A grande vantagem de usar IaaS é que os CSPs não precisam adquirir servidores, configurá-los, gerenciá-los e nem efetuar gastos com manutenção. Ou seja, a infraestrutura física é mantida e provida pelos CPs de forma que possam distribuir IaaS. No entanto, os CSPs são responsáveis pela segurança, desempenho e a confiabilidade dos CSs, através do monitoramento das cargas de trabalho que são executadas nas VMs contratadas.

De outra forma, PaaS simplificam a configuração e o gerenciamento dos recursos e do SaaS. Questões como a gerência de licenças de *software*, adequação do SaaS a plataformas específicas de *software* e a implantação do SaaS no ambiente de execução (uma VM) são menos complexas através de um PaaS, já que o CSP não necessita lidar com detalhes finos da execução e do desenvolvimento. Entretanto, PaaS podem ser considerados pouco flexíveis por exigir associações rígidas, tais como: o padrão ou a versão do software para execução ou desenvolvimento do SaaS; a dificuldade de controle da capacidade e da utilização de cada recurso (físico e virtual) e ainda por conta da dificuldade em estabelecer regras específicas de segurança e privacidade para os recursos.

Em geral, CPs não provêm um PaaS específico para seus consumidores, mas, um grupo de serviços que permite lidar com diferentes necessidades relativas ao desenvolvimento. Por exemplo, o *Amazon AWS* (<https://aws.amazon.com>) oferece um produto

particular que lida com o desenvolvimento e a implantação de SaaS: o *Amazon EC2* (<https://aws.amazon.com/pt/ec2/>). No entanto, outros serviços são necessários, tais como: o *Amazon S3* (<https://aws.amazon.com/pt/s3/>) para lidar com questões de armazenamento e o *Amazon CodeDeploy* (<https://aws.amazon.com/pt/codedeploy/>), útil para a implantação do SaaS.

Além dos modelos de CSs, o *National Institute of Standards and Technology* (NIST) pressupõe a existência de quatro principais modelos de implementação de nuvens computacionais: nuvens públicas, nuvens privadas, nuvens híbridas e nuvens comunitárias (Mell e Grance, 2011). Nuvens públicas comportam-se como silos de recursos computacionais públicos que podem ser explorados por Provedores de Serviços em Nuvem (CSPs) e que atendem a consumidores diversos em qualquer localização. Nuvens privadas, por sua vez, restringem-se a domínios particulares, atendendo a um grupo específico e que, em geral, obedecem a um conjunto homogêneo de restrições.

O modelo de nuvens híbridas decorre do fato de que os recursos físicos associados a uma nuvem são inevitavelmente finitos e que alguma tentativa de colaboração sempre tenderá a ocorrer. De fato, ao se conceber conceitualmente este modelo, (Group et al., 2013) reforça que a interconexão de nuvens computacionais é uma evolução inevitável. O último modelo sugerido por (Group et al., 2013) é designado como nuvens comunitárias, implicando no estabelecimento de um conjunto de distintas infraestruturas de nuvens computacionais coordenadas por uma das entidades participantes da comunidade, por um terceiro ou até mesmo por uma combinação de ambos.

Em geral, em nuvens públicas, cada CP disponibiliza imagens padrão de VMs e as comercializa. Contudo, essa é uma decisão que depende da escolha de uma boa estrutura de mapeamento de recursos físicos em virtuais (Pietri e Sakellariou, 2016), caso contrário, um inevitável desperdício pode ocorrer. É possível que sejam alcançados patamares de consumo dos recursos virtuais que extrapolam a capacidade física oferecida em cada *host*. Uma alternativa natural para manter a oferta constante de recursos é a integração de diferentes nuvens.

De acordo com (Durkee, 2010), a integração de nuvens computacionais é um movimento inevitável, no entanto, ainda caótico, pois as diversas combinações pressupõem diferentes possibilidades de coordenação e que ainda não foram claramente identificadas.

### 3.3 GERÊNCIA DE NUVENS COMPUTACIONAIS

Como uma nuvem provê a capacidade de atender a diferentes entidades e em distintos modelos de implementação, pode-se abordar o gerenciamento de nuvens em duas dimensões: em relação à entidade que interage com a nuvem e baseado no modelo de implementação.

Para a provisão de CSs, duas entidades podem necessitar interagir com a nuvem: CPs e CSPs. Enquanto CPs são os responsáveis por administrar a infraestrutura de uma nuvem, CSPs criam, mantêm e destroem CSs na infraestrutura oferecida por CPs. Ou seja, nuvens podem ser gerenciadas por CPs e CSPs, no entanto, cada entidade precisa possuir níveis de acesso distintos.

Em nuvens públicas, os Sistemas Operacionais de Nuvem (*Cloud Operating Systems* - COS), descritos por (Moreno-Vozmediano et al., 2012) como o componente chave de um DC, é responsável pela gerência da infraestrutura física e virtual, permitindo a orquestração e controle da provisão e do desenvolvimento de CS. Contudo, os consumidores podem optar por ter suas interações com a nuvem gerenciadas por outra ferramenta. De fato, CPs ou os provedores de IaaS, tais como: *Amazon AWS* (<https://aws.amazon.com/>), *Google Cloud Platform* (<https://cloud.google.com/>), *Microsoft Azure* (<https://azure.microsoft.com/>) e *IBM Cloud* (<https://cloud.ibm.com/>) implementam serviços que podem ser integrados e que contribuem na gerência das respectivas nuvens, cujos DCs estão sob seus domínios. A integração dos

diferentes serviços que contribuem para o gerenciamento de nuvens tem sido designada como Plataforma de Gerência de Nuvens (*Cloud Management Platform - CMP*). No entanto, conforme (Mainardi e Waite, 2018), CMPs têm sido empregados como um termo amplo, designando também desde um intermediário para CSPs gerenciarem seus serviços no domínio de um CP até mesmo possibilitando a integração de nuvens para o desenvolvimento de um CS.

A falta de padronização dificulta a identificação de interfaces e funcionalidades específicas de provedores de IaaS e CMPs. Por esse motivo, cada plataforma tem oferecido serviços distintos, provendo interfaces particulares e adotado diferentes estratégias para a integração dos serviços usados no gerenciamento. Nesta Seção são apresentados os esforços para concepção dos provedores de IaaS.

### 3.3.1 Provedores de IaaS

O dinamismo e a variedade de ambientes que nuvens computacionais podem proporcionar torna o gerenciamento dos recursos e serviços do ambiente uma tarefa que abrange várias necessidades. Apoiado sob essa constatação, o NIST apresenta em (Group et al., 2013) um esforço de padronização para nuvens. A arquitetura de referência sugerida pelo NIST permite identificar funcionalidades que potencialmente sejam encontradas em provedores de IaaS. No entanto, as atuais ofertas de mercado não indicam uma real conformidade. De fato, cada funcionalidade sugerida pelo NIST tem sido implementada sob interfaces distintas e através de diversos serviços que se integram de maneiras diferentes.

Conforme descrito na Figura 3.1, a arquitetura de referência de uma nuvem sugerida pelo NIST envolve quatro entidades: o consumidor, o CP, o *broker* de nuvem e um auditor. O *Broker* de Nuvem (*Cloud Broker - CB*) é identificado como uma entidade externa ao ambiente de um CP. Na verdade, conforme (A. Elhabbash et al., 2019), ela se destaca pela possibilidade de atender a diferentes necessidades. De acordo com (Tordsson et al., 2012), um CB deve, minimamente, cumprir duas funções: prover mecanismos de escalonamento que otimizem a alocação de VMs e oferecer uma interface uniforme de gerenciamento de VMs.

A Figura 3.1 permite identificar que a gerência de um CS envolve três funcionalidades: o suporte às tarefas de negócio; operações que permitem a configuração dos CSs e, por fim, operações que baseiam-se na portabilidade e na interoperabilidade para fornecer garantias de continuidade da execução de um CS.

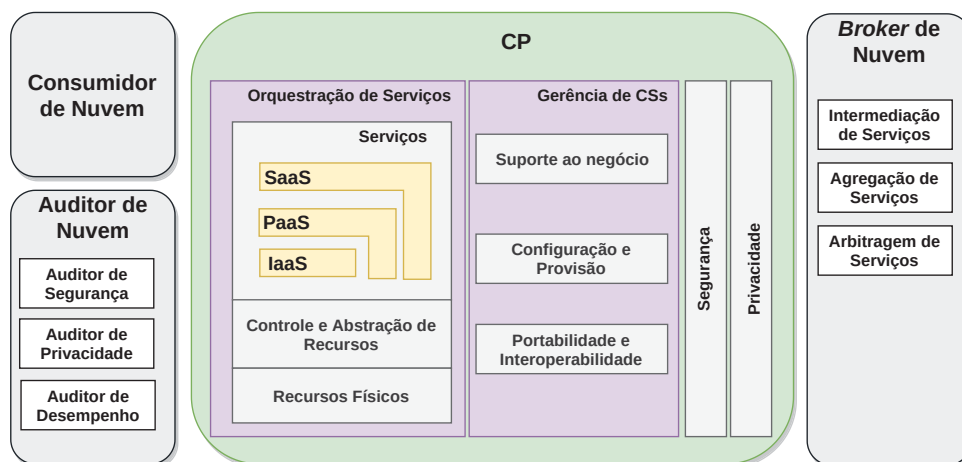


Figura 3.1: Arquitetura de referência para nuvens. Adaptado de (Group et al., 2013).

A portabilidade e a interoperabilidade são cruciais para a oferta sob demanda de recursos. Conforme o NIST, existem três preocupações que precisam ser abordadas: portabilidade de



dados; interoperabilidade de serviços e a portabilidade de sistemas. Quanto a portabilidade de dados, é esperado que consumidores possam transferir dados para e da nuvem, adequando a execução de suas cargas de trabalho conforme as necessidades identificadas externamente. A interoperabilidade de serviços diz respeito à necessidade de consumidores poderem executar suas cargas de trabalho em diferentes CPs, sem que para isso encontrem como obstáculo padrões proprietários ou ambientes de execução particulares. A portabilidade de sistemas diz respeito à capacidade de CSPs poderem migrar suas VMs (e respectivamente CSs executados nessas VMs). O NIST, contudo ressalta, que um CP deve estar apto a migrar tanto cargas de trabalho particulares como serviços, ou seja, todas as VMs que estejam associadas ao CS. Ainda, o NIST sugere que seja prevista a capacidade de migração tanto de instâncias de uma VM como também de imagens.

É possível depreender que a portabilidade ou a migração apresentem diferentes nuances. Quando se trata da migração de instâncias de VMs é possível considerar duas situações: a migração de uma VM específica cuja transferência reequilibrará um *host* como também considerar que se trate da migração de todas as VMs que compõem um CS em um determinado momento. Quando se trata da imagem de uma VM, é possível depreender que o propósito seja o de reaproveitar padrões de VMs particulares entre CPs, uma prática que potencializa a interoperabilidade.

Embora o NIST sugira funcionalidades para lidar com a portabilidade e a interoperabilidade, não existem referências específicas de como lidar com a migração. Talvez, por esse motivo, Provedores de IaaS adotam estratégias diferentes. Por exemplo, o *Amazon AWS* provê seis serviços distintos na sua plataforma de gerenciamento, cujas funcionalidades estão associadas à detecção e ao monitoramento de condições que demandem uma migração, no entanto, o planejamento e a execução da migração não são efetuados por um serviço *Amazon*. Em geral, o planejamento da migração é delegado a um serviço externo, exigindo um esforço de integração pelo operador do ambiente.

A *Google Cloud Platform - GCP* oferece soluções nativas para migração (Woudenberg, 2016), usualmente disparadas em decorrência de necessidades de manutenção detectadas no ambiente. O serviço *Google Compute Engine - GCE* (<https://cloud.google.com/compute/>) provê estratégias para detectar eventos relativos a manutenção não programada tais como: atualizações necessárias de *hardware* e *software*, manutenção de rede elétrica e falhas de *hardware*. Diferente do *Amazon AWS*, o planejamento de migrações é organizado e executado pelo GCE.

É possível constatar que há um papel relevante dos CBs no gerenciamento de nuvens e que as CMPs têm soluções distintas para a integração dos diversos serviços que podem contribuir nas ações de gerenciamento. Embora esforços de padronização de CMPs já tenham sido apresentados (Council, 2017), há diferentes abordagens de integração, inclusive com distintas finalidades para um CMP (A. Elhabbash et al., 2019).

### 3.4 GERÊNCIA DE RECURSOS EM NUVENS COMPUTACIONAIS

De acordo com (Jennings e Stadler, 2015), o gerenciamento de recursos em nuvens computacionais é compreendido como o processo de alocar capacidade de processamento, de armazenamento, de largura de banda e (indiretamente) de energia para cada CS. Tal alocação precisa atender a objetivos de desempenho já estabelecidos, tais como: os relacionados ao desempenho dos CSs, à infraestrutura ofertada pelo CP e os relativos aos consumidores dos CSs. Entretanto, as técnicas tradicionais de gerenciamento de recursos não são adequadas para Computação em Nuvem, pois da virtualização decorrem requisitos que até então não eram considerados, tais como a elasticidade e o multi-inquilinato (*multi-tenancy*) (Parikh, 2013).

A limitação do domínio de oferta de recursos afeta a provisão de CSs, ocasionando, em casos extremos, a interrupção da oferta do CS. Neste caso, consumidores e CSPs são afetados e, penalidades previstas em acordos podem causar prejuízos tanto a CSPs como CPs. Com o propósito de diminuir os prejuízos decorrentes da limitada oferta de recursos para CSs, a migração de VMs tem sido utilizada como a estratégia para provisão elástica de recursos.

Nuvens possuem uma intrínseca distribuição de responsabilidades entre CPs e consumidores (incluindo os CSPs). Recursos físicos são gerenciados por CPs, no entanto, recursos virtuais são diretamente afetados pelas políticas, restrições e objetivos de consumidores. Conforme (Bittencourt et al., 2015), cada modelo de serviço demanda, do CP, distintas capacidades de gerenciamento, no entanto, flexibilizando o controle dos recursos em outros níveis, como descrito na Figura 3.2.

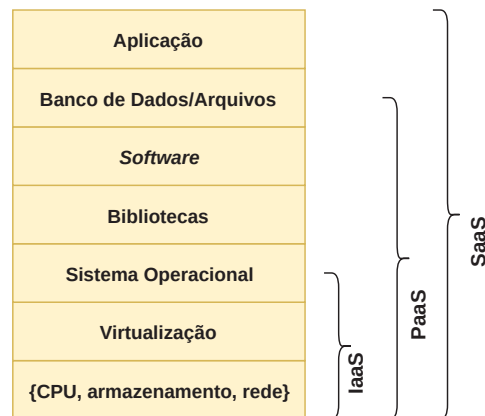


Figura 3.2: Níveis de gerenciamento com interferência direta do CP. Adaptado de (Bittencourt et al., 2015).

Os recursos de *hardware* assim como os recursos virtuais e as operações executadas pelo Sistema Operacional somente podem ser controlados pelos CPs em serviços baseados no modelo IaaS. Contudo, consumidores têm acesso a bibliotecas, arquivos, bancos de dados e aos serviços que têm como alvo os recursos virtualizados. Serviços baseados no modelo PaaS têm um controle mais amplo do CP, delegando ao consumidor apenas o produto desenvolvido como CS. E, por fim, serviços baseados em SaaS somente podem ser executados pelos consumidores, de forma que tanto em quais VMs e *hosts* estão hospedados os CSs é totalmente desconhecido pelo consumidor.

### 3.4.1 Funcionalidades na Gerência de Recursos

Um conjunto de funcionalidades necessárias para ambientes de nuvens foi proposto em (Jennings e Stadler, 2015). Conforme descrito na Figura 3.3, a cada entidade são associadas distintas responsabilidades.

As responsabilidades associadas aos CPs incluem: o escalonamento de recursos; a elaboração de perfis de uso; o monitoramento de recursos físicos e virtuais e o cálculo de preços. O escalonamento de recursos visa atender às necessidades globais do ambiente seja na alocação inicial como durante o tempo em que a VM mantém-se atendendo a cargas de trabalho. No entanto, também atua coordenando o mapeamento de recursos virtuais nos respectivos físicos, ou seja, controlando a ação de *hipervisores*, executando ajustes que podem influenciar na oferta de recursos no ambiente.

A elaboração de perfis que representem a demanda dos recursos exige que se equilibre entre ações corretivas e preventivas. Por esse motivo, são esperados perfis baseados em modelos que permitem descrever tanto o que se prevê de uso dos recursos como a real utilização em um

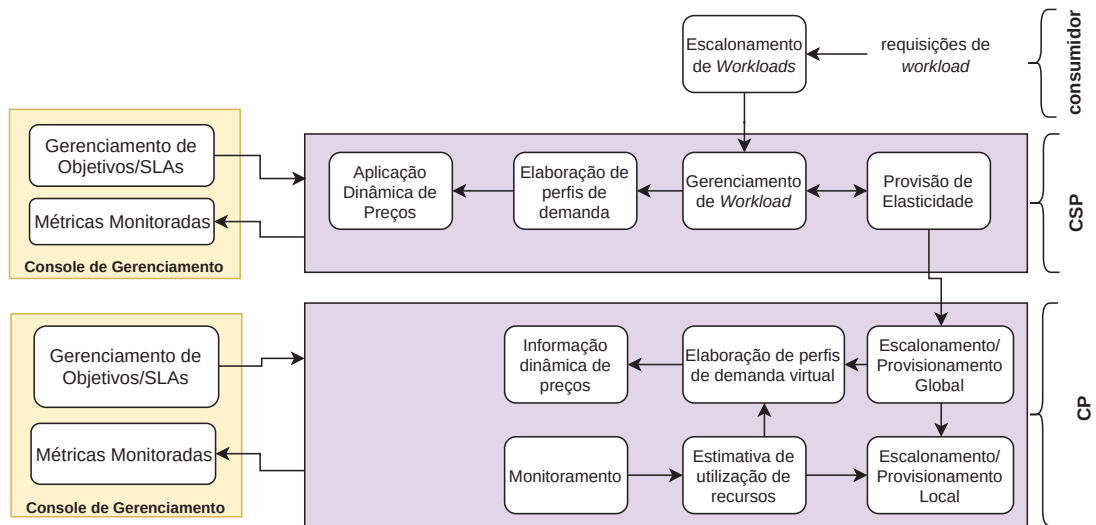


Figura 3.3: Funcionalidades na gerência de recursos em nuvens, conforme (Jennings e Stadler, 2015).

determinado momento. Para ambos os casos, diferentes tecnologias podem ser empregadas, variando de técnicas baseadas em estatística e probabilidade às que se baseiam em algoritmos e modelos da Computação Evolucionária.

Enquanto CPs se responsabilizam pelo monitoramento e o controle de recursos físicos do ambiente, CSPs controlam seus CSs. Por essa razão, CSPs dedicam-se às seguintes funcionalidades, descritas na Figura 3.3: provisão de elasticidade; gerenciamento de cargas de trabalho; elaboração de perfis de demanda e aplicação de preços.

Uma característica peculiar que nuvens oferecem aos CSs é a capacidade elástica de uso de recursos. É responsabilidade do CSP, o controle da provisão dos CSs e isso inclui disparar ações para garantia da elasticidade. As variações de demanda por recursos podem exigir diferentes estratégias tais como: redimensionamento da oferta de recursos físicos à VM; migração da carga de trabalho para outra VM e a migração de uma VM. A primeira estratégia é a menos utilizada, pois o redimensionamento de VMs exige modificações em contratos, uma decisão, em geral, onerosa para ambas as partes. A migração de cargas trabalho envolve um monitoramento de cada carga de trabalho desde o instante em que a VM inicia sua execução até o encerramento. Esta é uma operação que dificilmente apresenta boa escalabilidade, devido à possibilidade da geração de muitas cargas de trabalho em pouco tempo. A migração de VMs é o resultado do monitoramento ao nível de *hosts*, tipicamente baseada, como mencionado anteriormente, na detecção de *thresholds*.

### 3.4.2 Objetivos na Gerência de Recursos

Os atores de um ambiente de nuvem têm interesses diferentes, influenciando nos objetivos de gerenciamento dos recursos que se pretende alcançar (Jennings e Stadler, 2015). Embora possam ser conflitantes, Pietri e Sakellariou (2016) destacam a existência de três categorias de objetivos que visam otimizar o gerenciamento de recursos: relacionados à utilização de recursos; associados às questões monetárias e para o controle do consumo de energia.

Objetivos de gerenciamento somente podem ser alcançados se ações forem previstas. Conforme descrito por Mishra et al. (2012); Marinescu (2013), para lidar com as três categorias de objetivos, deve-se considerar cinco ações de gerenciamento: a detecção e a mitigação de *hot/cold spots*, a consolidação de *hosts*, o balanceamento de carga, o controle de admissão e a garantia de QoS.



Se é detectado que um *host* ultrapassa um limite considerado seguro na provisão de um ou mais recursos virtuais, então uma ou mais VMs podem ser consideradas o prenúncio de um desequilíbrio no *host*. Esse fenômeno é conhecido como *hot spot* e diminuir a sua ocorrência exige readequações rápidas no mapeamento dos recursos físicos. A ocorrência da subutilização da capacidade dos recursos físicos de um *host* denota um *cold spot* que também pode exigir um remapeamento dos recursos.

Quando se reconhece que existem *hosts* subutilizados em um ambiente, pode-se optar pela consolidação de servidores, ou seja, a diminuição da quantidade de *hosts* em que o consumo de recursos é considerada baixa. Para atingir o propósito, grupos de VMs de *hosts* considerados subutilizados são migradas para outros *hosts*, permitindo que o processamento das cargas de trabalho seja concentrado em um menor número de *hosts* ativos (Vogels, 2008).

O balanceamento de carga tem um propósito diferente da consolidação, estando associado à necessidade de equilibrar a quantidade de *hosts* sobrecarregados. Como resultado, tem-se grupos de VMs migrando de *hosts* sobrecarregados para outros que ainda suportem novas VMs e suas respectivas cargas de trabalho. Em geral, o balanceamento de carga reduz a concentração do consumo de recursos pela detecção de discrepâncias, permitindo que a capacidade de execução das cargas de trabalho no DC atinjam um nível de equilíbrio (Mishra e Sahoo, 2011).

O balanceamento de carga pode contribuir no gerenciamento de energia, evitando a sobrecarga de *hosts* que, inevitavelmente, gastariam mais energia e também dissipariam mais calor que o esperado. Um propósito semelhante pode ser direcionado à consolidação de servidores, impedindo que *hosts* subutilizados continuem consumindo energia. No entanto, para se atender a ambos os objetivos, tipicamente recorre-se a inspeções periódicas de *hot/cold spots*.

### 3.5 GERÊNCIA DE RECURSOS BASEADA NA MIGRAÇÃO DE VMS

A virtualização proporciona maior flexibilidade aos CSs, contribuindo para que os recursos de um ambiente de nuvem possam ser melhor compartilhados e, conseqüentemente, para que haja maior disponibilidade. Contudo, o intrínseco dinamismo nas demandas por recursos estabelece um desafio: como provê-los dinamicamente? Nesse contexto, a migração de VMs é a principal estratégia para o gerenciamento de recursos em um ambiente de nuvem, pelo potencial abrangente de oferta de recursos, ou seja, por ter a capacidade de reaproveitamento de recursos e também por contribuir no reequilíbrio de *hosts*.

As cargas de trabalho executadas nas VMs possuem comportamento incerto, podendo variar a demanda por recursos inesperadamente. Localizar recursos em um ambiente de nuvem e efetuar novos mapeamentos é uma tarefa que exige que planos de migrações sejam prontamente construídos. Contudo, os dados que permitem a construção de tais planos nem sempre estão disponíveis ou atualizados.

Conforme descrito por Wood et al. (2007), os dados que promovem o planejamento de migrações podem provir de duas fontes, interna e externa. No primeiro caso, o hipervisor é a principal fonte, fornecendo dados sobre o consumo de recursos pela VM. Equipamentos de rede também podem ser consideradas fonte internas, por oferecer informações sobre o tráfego das VMs em seus respectivos *hosts*. Fontes externas são aquelas que provêm informação que pode ser constatada externamente por quaisquer outras entidades, tais como: objetivos e restrições em SLAs, dados fornecidos por ferramentas de gerenciamento, estatísticas fornecidas pelo CP e ainda dados estatísticos fornecidos pelas aplicações e Sistemas Operacionais.

Ambas as categorias de dados estão baseadas em distintos componentes que, usualmente, dependem de intervenção humana. Isso ocorre, por exemplo, na captura de informações específicas de VMs, exigindo que um operador localize dados sobre o uso de cada recurso, para cada VM.

Esse é um panorama, inevitavelmente, propenso a falhas. Além disso, é razoável considerar que os dados coletados podem sofrer com alguma desatualização, requerendo exaustivas verificações.

Embora a migração de VMs seja considerada a estratégia mais adequada para o gerenciamento de recursos em nuvem (Mishra et al., 2012), pode-se constatar que poucas iniciativas foram concebidas com o propósito de integração das diferentes fontes de dados que contribuem para o planejamento de migrações. Tem-se lidado com o ciclo de vida de uma migração como uma tarefa à parte no gerenciamento de recursos, tendo sido negligenciadas ações que, efetivamente, contribuem para a construção de planos de migração.

É possível constatar que a ausência de ações de gerenciamento dedicadas às migrações contribui para as seguintes situações: a distribuição da informação dependendo sobretudo de um operador; os dados sobre os recursos, usados no planejamento das migrações, dificilmente estão atualizados e os objetivos e as restrições (usualmente identificadas em SLAs) de cada CS, *host*, VM e *workload* dificilmente são respeitados. Todas essas situações contribuem para planos de migração que não sejam efetivos, ou seja, que ao invés de gerar melhor distribuição dos recursos pode agravar ainda mais o desequilíbrio.

A gerência da migração de VMs é um conjunto de ações que permite a construção dinâmica de planos de migração em resposta a demandas de recursos. Tem-se como principal propósito a redução da interferência humana, permitindo que as diferentes fontes de dados que contribuem para o planejamento da migração possam contribuir na gerência dinâmica dos recursos de um ambiente de nuvem.

### 3.6 CONSIDERAÇÕES FINAIS

VMs são a unidade básica que representa os recursos disponibilizados por *hosts* que são parte de DCs e que, por sua vez, constituem nuvens computacionais. Diferentes modelos de serviços e de implantação de nuvens determinam a maneira como os CSs são desenvolvidos e entregue às diferentes entidades que interagem com nuvens (consumidores, CSPs e CPs).

A migração de nuvens é uma operação natural, decorrente do comportamento dinâmico das cargas de trabalho que são executadas nas VMs. Em decorrência das propriedades de elasticidade e multi-inquilinato, o uso dos recursos virtuais das VMs pode variar, assumindo, inclusive patamares que não podem ser suportados pelos recursos físicos dos *hosts*, e demandando, portanto a necessidade de uma migração.

A gerência de nuvens tem sido associada a plataformas designadas CMPs. Em geral, tais plataformas são dedicadas ao gerenciamento de recursos e serviços do tipo IaaS. Especialmente no tocante ao gerenciamento de recursos, funcionalidades mínimas podem ser identificadas e suas interações descritas.

A migração de VMs é um expediente natural para o gerenciamento de recursos, demandando que cada etapa do ciclo de vida de uma migração adote técnicas e estratégias que atendam aos propósitos de gerenciamento da nuvem.

## 4 ESTADO DA ARTE EM GERÊNCIA DE MIGRAÇÕES

Neste capítulo são apresentados trabalhos que têm como propósito prover ações para o gerenciamento da migração de VMs. Tais ações envolvem tanto o planejamento como a execução de uma migração. Dois cenários de migração são considerados: intra-DC e inter-DC. No primeiro cenário estão as migrações que ocorrem no âmbito de um único ambiente de nuvem e especificamente nos *hosts* associados a um único DC. No segundo cenário estão as migrações que podem ocorrer entre diferentes DCs. Para o cenário inter-DC pode-se considerar duas possibilidades: os DCs envolvidos na migração estarem em um mesmo ambiente de nuvem e os DCs serem localizados em ambientes de nuvem distintos. Nesta tese focamos apenas no cenário inter-DC de um ambiente único de nuvem.

A Seção 4.1 permite contextualizar que focos têm sido a preocupação dos trabalhos que visam lidar com o gerenciamento da migração de VMs. A Seção 4.2 apresenta trabalhos que lidam com a migração no cenário intra-DC. A Seção 4.3 apresenta os esforços para gerenciar migrações no âmbito inter-DC. Dos trabalhos apresentados nas seções 4.2 e 4.3 foi possível constatar os desafios que estão associados à gerência de migrações, sendo apresentados na Seção 4.4. Ao final do capítulo, na Seção 4.5, ressaltamos a necessidade de se conceber uma abordagem de gerência de migrações de VMs.

### 4.1 INTRODUÇÃO

Sistemas Operacionais de Nuvens (*Cloud Operating Systems* - COS) e CMPs são as típicas entidades que lidam com o gerenciamento de recursos em nuvens e, consequentemente, em DCs. Diferentes dimensões permitem medir a abrangência de um DC. É possível que seja necessário gerenciar de centenas a milhares de *hosts* cujos recursos são virtualizados a fim de diminuir a complexidade de lidar com diferentes inquilinos (*tenants*). DCs também podem operar em um domínio privado, assim como terem sido concebidos para atender a consumidores publicamente. No primeiro caso, o comportamento dos *tenants* tende a ser mais previsível, no entanto, em nuvens públicas, o comportamento é imprevisível.

A capacidade de um DC em lidar com flutuações persistentes em *workloads* depende de sua habilidade de migrar VMs. Muitas pesquisas já foram apresentadas, descrevendo estratégias para lidar com a reconfiguração dinâmica de recursos em um DC através de migrações de VMs. Tais estratégias incluem esforços para se atingir diferentes objetivos: consolidação de servidores, balanceamento de carga, recuperação de falhas e diminuição de violações de SLA. Basicamente, essas estratégias têm focado em dois esforços: na idealização de arquiteturas que contribuem na elaboração de planos de migração e em mecanismos que otimizem a execução das migrações. Trabalhos que se enquadram em ambas as categorias são apresentadas neste capítulo.

### 4.2 GERÊNCIA DE MIGRAÇÕES INTRA-DC

Embora um DC seja parte de um ambiente de nuvem, os trabalhos relacionados à gerência intra-DC focam em soluções de gerenciamento no âmbito de um único DC. Nesta Seção são apresentados trabalhos que buscam automatizar a tarefa de planejar migrações.

#### 4.2.1 (Wood et al., 2007)

Até onde pode-se analisar, a primeira abordagem de gerência da migração de VMs foi apresentada em (Wood et al., 2007). Nesta abordagem, todas as migrações ocorrem entre *hosts* baseados no mesmo hipervisor em um domínio intra-DC e as decisões são tomadas no Plano de Controle (*Control Plane - CP*). Três entidades pertencem ao CP: o *Profiling Engine* (PE), o *Hotspot Detector* (HD), e o *Migration Manager* (MM) (Figura 4.1). Cada entidade no CP implementa políticas distintas visando um objetivo principal: eliminar a ocorrência de *hot spots* no DC.

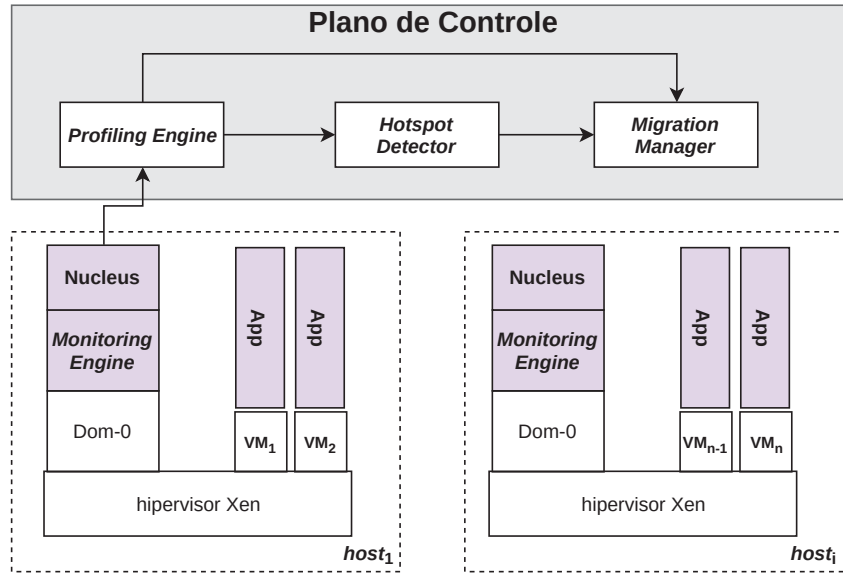


Figura 4.1: Automação da Migração de VMs, proposto por Wood et al. (2007)

Cada *host* possui uma entidade local designada *Nucleus* que continuamente coleta dados, sob duas perspectivas: dados de VMs que podem ser externamente acessíveis e os dados internos de VMs. Ambos os conjuntos de dados são, posteriormente, enviados ao PE. O trabalho de Wood et al. (2007) avalia se informações de VMs que são acessíveis externamente às VMs são suficientes para a construção de planos de migração eficientes. Ou seja, é analisado se é possível ignorar informações relativas aos *workloads* e ao CS em execução em uma VM. Foi constatado que ambos os domínios de informação não devem ser ignorados.

Variações no uso dos recursos podem ocorrer durante as coletas, demandando a adoção de estratégias que diminuam a ocorrência de *thrashing* de migrações, ou seja, o disparo constante de migrações em decorrência de mau planejamento (Liu et al., 2020). Por essa razão, os perfis de uso dos recursos são elaborados em janelas deslizantes de tempo que são organizadas em intervalos. Ao final, tem-se várias distribuições de probabilidade e séries temporais, ambas usadas nas decisões das etapas seguintes.

Para comprovar a sobrecarga de um *host* ou de uma VM, uma métrica designada *Volume(V)*, descrita na Equação 4.1, é utilizada para confirmar se o uso dos recursos excedem um limiar (*threshold*) para o caso de um *host* ou a capacidade contratada para uma VM. O volume de uma VM ( $V(m)$ ) é usado para ordenar VMs que têm o potencial de se tornarem migrantes. Por outro lado, o volume de um *host* ( $V(h)$ ) pode ser usado para classificar *hosts* pela ordem de uso dos recursos e, definir, por exemplo, uma lista de potenciais destinos para uma VM.

$$V = \frac{1}{1 - U(cpu)} \times \frac{1}{1 - U(ram)} \times \frac{1}{1 - U(bw)} \quad (4.1)$$

Na equação 4.1, tem-se que  $cpu$ ,  $ram$ , e  $bw$  são, respectivamente, a capacidade de processamento, de memória e dos recursos de rede e  $U(r)$  denota o atual uso do recurso  $r$  por uma VM ou por um *host*.

$V$  é usado para estabelecer uma métrica que captura o uso de todos os recursos por uma VM, designada Razão Volume-Tamanho (*Volume-Size Ratio* - VSR). VSR é a razão entre o volume  $V$  do uso de recursos de uma VM  $m$  ( $V(m)$ ) e o tamanho dessa VM, designado pelo uso de memória em um determinado instante, ou seja  $S = U(m, ram)$ . Essa métrica é usada para classificar as VMs que têm o potencial de migrar.

$$VSR = \frac{V}{S} \quad (4.2)$$

A detecção de um *hot spot* requer a escolha de VMs migrantes. Um algoritmo guloso determina como VM migrante a que possuir maior VSR para cada *host* sobrecarregado. O maior VSR indica maior capacidade de transportar o maior volume ( $V$ ) de carga de uma VM por unidade de memória transportada. Como destino são sempre escolhidos os *hosts* de menor volume. Contudo, se esses destinos não puderem atender à demanda prevista da VM, outro *host* menos sobrecarregado é escolhido. Essa ação é coordenada pelo MM, sendo iniciada apenas quando todos mapeamentos tiverem sido definidos e encerrada quando nenhum *host* estiver operando recursos acima de *thresholds* pré-definidos.

Para cada VM da lista ordenada pelos respectivos VSRs são identificados *hosts* que possam atender à capacidade esperada para cada recurso. A lista de *hosts* candidatos para cada VM migrante é, posteriormente, ordenada pelos respectivos volumes  $V$  de cada *host*, gerando portanto, uma lista de várias tuplas  $\{m, < h_1, h_2, \dots, h_n >\}$ . Ou seja, para cada VM  $m$  tem-se um conjunto ordenado de diferentes *hosts* candidatos.

Conforme (Liu et al., 2020), ocorre *thrashing* de migrações quando uma mesma VM precisa sucessivamente migrar entre diferentes *hosts*, denotando, portanto, que as migrações não foram bem planejadas, já que o destino escolhido sempre é um *host* instável. Adotar um *threshold* fixo tende a tornar o planejamento das migrações vulnerável a *thrashings* (Luiz et al., 2010). Além disso, a escolha de um valor relaciona-se muito ao propósito da operação. Valores maiores para *thresholds* são adequados para detecções menos agressivas de *hot spots* e podem não ser adequadas para lidar com outros potenciais gatilhos para avaliação de migrações, tais como: violações de SLA, subutilização de recursos e contenção de recursos. Além disso, no trabalho de Wood et al. (2007), as migrações em curso não são claramente consideradas nem na definição de uma janela deslizante de tempo e nem no cálculo do volume seja da VM migrante como dos *hosts* de origem e destino. Desconsiderá-las também motiva a produção de *thrashings* de migração.

Dentre as limitações verificadas no trabalho de Wood et al. (2007) pode-se identificar: a restrição da proposta à detecção de apenas uma condição de migração (*hot spots*); o modelo foi concebido no âmbito de um domínio identificado como intra-DC, mas, não são esclarecidas situações típicas de um DC tais como a interoperabilidade de diferentes hipervisores; o controle de migrações em curso não fica claramente evidenciada, ou seja, não existem mecanismos que acompanhem o efetivo dos recursos durante migrações, impedindo que tais dados participem de decisões mais precisas de detecção e erradicação de *hot spots*.



#### 4.2.2 (Xu e Fortes, 2011)

No trabalho de Xu e Fortes (2011), migrações são planejadas de acordo com a primeira ocorrência de uma das três condições: emergência térmica, baixa eficiência energética e contenção de recursos. Para todos os casos, conforme descrito na Figura 4.2, sensores fazem a coleta periódica dos dados reportando a um Controlador Global (*Global Controller - GC*). Como existe mais de uma condição a ser observada, uma ordem de prioridades deve ser estabelecida e a identificação de uma condição dispara uma *thread* de ações para o planejamento. Enquanto isso, o monitoramento de outros *hosts* pode continuar a ocorrer.

Para Xu e Fortes (2011), migrações demandam dados de duas camadas: infraestrutura e virtualização, sendo, usualmente tratados de forma isolada. Por essa razão, apresenta uma abordagem designada como camadas cruzadas (ou, do inglês *cross-layer*). Diferente da solução apresentada por Wood et al. (2007), os dados são coletados e analisados ao mesmo tempo, por meio do suporte de sensores.

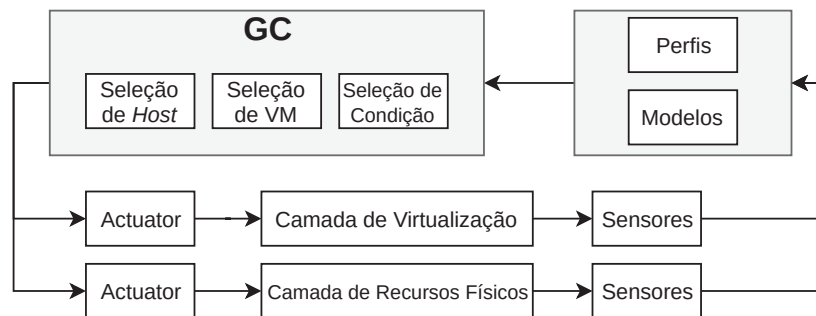


Figura 4.2: Controle da Migração Multi-Camadas, conforme (Xu e Fortes, 2011).

Para evitar que picos espúrios no uso dos recursos influenciem em decisões equivocadas de migrações, no trabalho de Xu e Fortes (2011), os dados usados na detecção de um gatilho de migração são coletados em intervalos definidos por meio de janelas deslizantes. Em cada janela ( $W$ ), dispara-se um pré-alarme, caso seja identificado que a quantidade de recursos violando um valor de *threshold* ultrapasse um valor determinado pelo administrador do ambiente. Contudo, outra medida é adotada para evitar que a decisão de migrar gere prejuízos. Através de uma análise de tendência, é verificado se o atual uso de recursos da VM mantém, num futuro próximo, o comportamento de sobrecarga. Ou seja, se a análise de tendência indicar que continuará ocorrendo violação de *threshold*, então uma migração é demandada. Ou seja, se ao final da verificação da condição de migração, forem disparados dois alarmes, então inicia-se o planejamento de uma migração.

A definição do tamanho de uma janela deslizante (ou o intervalo de tempo em que ela se mantém) é discutido por Xu e Fortes (2011). Uma janela deslizante pequena produz um esquema agressivo de detecção, com maior potencial de geração de falsos positivos. Janelas deslizantes maiores forçam que as detecções sejam mais lentas, possibilitando mais detecções de eventos transientes e migrações desnecessárias. Para dirimir o tamanho adequado, Xu e Fortes (2011) sugerem que janelas deslizantes coincidam com a duração em que um *host* ou uma VM mantêm-se com a mesma configuração de uso dos recursos. Esse intervalo de tempo é descrito como o Tempo Estável ( $T_{stable}$ ). O intervalo de tempo que *host* e VMs mantêm-se estáveis é o tamanho da janela deslizante, sendo necessário que ela seja superior ao tempo usado para a execução de uma migração ( $T_{migration}$ ), ou seja:  $T_{stable} = W > T_{migration}$ . Para calcular  $T_{stable}$ , foi feita uma aproximação, VMs com 512M migrariam em até 40s, sendo o limite da duração de uma  $W$ .

VMs migrantes são selecionadas conforme a condição detectada. Em caso de emergência térmica, o objetivo é diminuir ao máximo a sobrecarga de processamento dos *hosts* que operam acima dos *thresholds*. A razão entre uso de CPU e tamanho de memória da VM (*Utilization to Memory Size* - *USR*) é usada para classificá-las em ordem decrescente. Os maiores valores de *USR* são candidatos à migração. Em caso de contenção de recursos, o GC identifica que o motivo da sobrecarga é a concentração de uso em determinado recurso, o que causa uma disputa constante entre diferentes VMs, com consequentes violações de SLA. Para cada VM é identificado o uso do recurso escasso e calculada uma média ( $M_{down}$ ). As VMs com uso acima da média são candidatas à migração. Em caso de baixa eficiência energética, a solução adotada é a consolidação dos servidores, migrando-se todas as VMs do *host* que consome mais energia do que era esperado.

No trabalho de Xu e Fortes (2011), as condições que levam a uma migração são priorizadas pelo administrador do DC, o que permite considerar ser possível uma migração ocorrer por uma razão que não seja a efetiva motivação do desequilíbrio na oferta de recursos. Ainda, essa abordagem também tende a elevar a capacidade de processamento requerida, uma vez que todas as condições precisam ser verificadas. Outro aspecto limitante é o potencial para *thrashings* de migração, já que uma escolha não adequada pode contribuir ainda mais para o desequilíbrio que já existia.

#### 4.2.3 (Beloglazov e Buyya, 2012)

O foco do trabalho de Beloglazov e Buyya (2012) é a definição de uma estratégia que otimize o *trade-off* entre solucionar *hot spots* e continuar a atender aos diferentes *workloads* em operação. Os autores observaram que esquemas agressivos de migração, ou seja, com migrações que ocorrem em intervalos de tempo curtos tendem a produzir mais violações de SLA. Por esta razão, Beloglazov e Buyya (2012) constataram que o intervalo entre migrações deveria ser o maior possível, evitando assim que mais violações de SLA fossem produzidas, inclusive influenciando nos resultados dos *workloads* em execução.

Tendo verificado que as estratégias anteriores para a detecção de gatilhos de migração são limitadas, é proposta uma nova heurística. Além de possibilitar lidar com as garantias de QoS, a nova heurística também responde de forma dinâmica à necessidade de consolidação de servidores. Em (Beloglazov e Buyya, 2013), é descrito que os perfis de uso dos recursos são coletados em janelas deslizantes de tamanho variável e o comportamento entre migrações é otimizado por estados implementados por uma Cadeia de Markov. Outro aspecto relevante é a adoção de valores adaptativos de *thresholds*, constantemente reajustados em consonância com o incremento do uso de CPU em relação ao que foi observado anteriormente.

Beloglazov e Buyya (2012) destacam três políticas para a seleção de VMs migrantes: Política do Tempo Mínimo de Migração (*Minimim Migration Time Policy* - MMT), Escolha Aleatória e Política da Máxima Correlação (*Maximum Correlation Policy* - MC). Todas são aplicadas iterativamente sempre que um *host* é considerado sobrecarregado. Após a seleção de uma VM migrante, o *host* novamente é checado e se continuar sobrecarregado, as políticas são aplicadas novamente para a busca de outra VM migrante. No trabalho de Beloglazov e Buyya (2012), a consolidação é um propósito que visa atender a eficiência energética, por esse motivo, a seleção de um destino ocorre em dois passos: ordenação das VMs migrantes em ordem decrescente de uso da CPU e identificação do *host* que provê o menor aumento no consumo de energia.

A avaliação das propostas de Beloglazov e Buyya (2012) é executada sob uma arquitetura de DCs baseada em uma entidade central, descrita na Figura 4.3. Considera-se que o ambiente seja um DC de larga escala, contendo uma quantidade de  $n$  *hosts* heterogêneos. Um Gerenciador

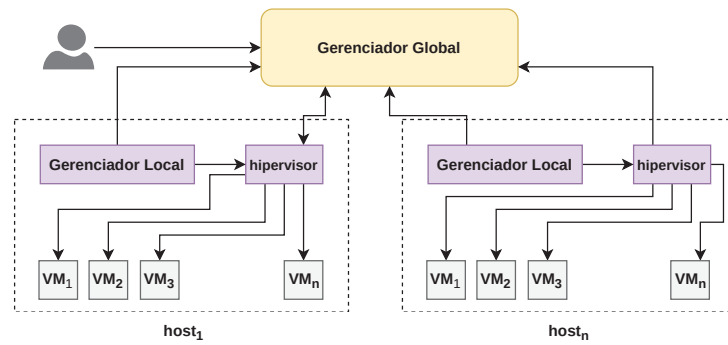


Figura 4.3: Controle da Migração Multi-Camadas, conforme (Beloglazov e Buyya, 2012).

Global executa coletas periódicas dos dados que cada Gerenciador Local informa. O Gerenciador Global estabelece dois *thresholds*: superior e inferior. O limite superior é utilizado para identificar *hosts* sobrecarregados e o inferior para determinar se as VMs de um *host* devem ser migradas, consolidando outros servidores. As migrações são planejadas pelo Gerenciador Global que pode, a cada migração, atualizar os valores dos *thresholds*.

#### 4.2.4 Outros trabalhos

O foco do trabalho de Baccarelli et al. (2015) é a execução de planos de migração que não contraiam prejuízos, especificamente, o aumento do consumo de energia. Por esse motivo, um modelo que permita monitorar o uso da largura de banda de rede durante a execução de uma migração é apresentado. Nesse modelo, são consideradas quatro restrições: o uso máximo de memória tolerada durante a migração; quantidade de vezes que uma operação *stop-and-copy* é executada; número de rodadas necessárias para a conclusão da migração e a largura de banda máxima que pode ser utilizada. Tais restrições permitem a definição de um Problema de Otimização Multi-Objetivo (*Multi-Objective Optimization Problem* - MOP), designado Problema de Otimização de Largura de Banda (*Bandwidth Multi-Objective Optimization Problem* - BMOP).

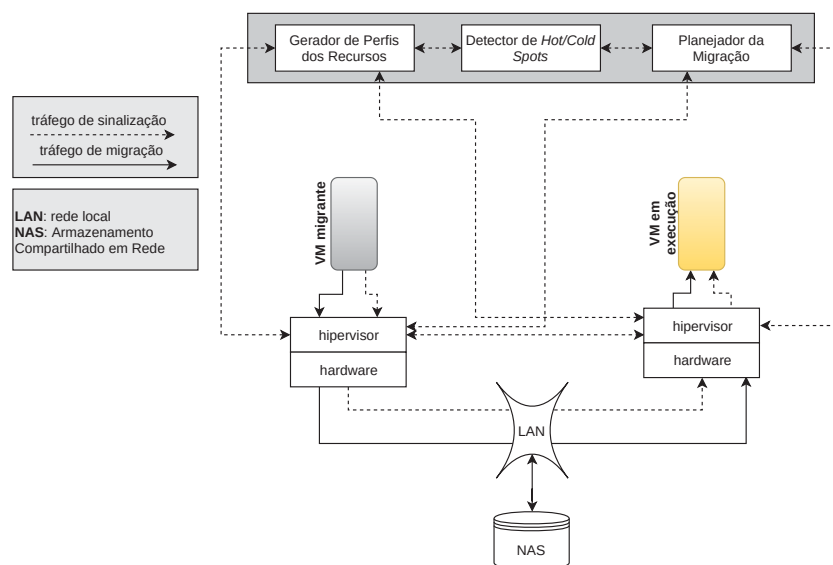


Figura 4.4: Arquitetura de referência para um DC virtualizado, conforme (Baccarelli et al., 2015)

Planos de migração são elaborados sob uma arquitetura de DC, considerada por Baccarelli et al. (2015) como uma arquitetura de referência (Figura 4.4). Um *Migration Planner*



(MP) controla as atividades de planejamento, com o suporte de duas entidades: o Gerenciador de Perfis de Uso dos Recursos (*Resource Profiler* - RP) que coleta dados sobre o uso de recursos; o Detector de *Hot/Cold Spots* (*Hot/Cold Spot Detector* - HCD) e o Planejador de Migrações (*Migration Planner* -MP) que reage às sinalizações que provenham do HCD. A possibilidade de ocorrências *Hot* e *Cold Spots* é avaliada periodicamente, através de informações coletadas pelo RP.

Um Gerenciador de Largura de Banda (*Bandwidth Manager* - BM) é o principal componente da solução de Baccarelli et al. (2015). Seu propósito é conduzir a execução de um plano de migração já construído, requerendo, portanto, informações que, usualmente, são fornecidas pelo hipervisor, tais como: o tamanho de memória requerida pela VM; o tamanho de largura de banda de rede alocada para a VM e a taxa de páginas sujas de memória deixadas pela VM. Essas informações são coletadas a partir do momento em que migração inicia, exigindo do BM ações de coleta tanto no *host* de origem como no destino.

Os experimentos conduzidos pelos autores demonstram que a quantidade de vezes em que uma operação *stop-and-copy* foi necessária diminuiu em comparação a um cenário em que BM não foi utilizado. Ou seja, a geração de páginas sujas de memória diminuiu e, portanto, exigindo menos tráfego extra para o transporte de tais páginas de memória. No entanto, é importante ressaltar que a abordagem é baseada eminentemente em informações coletadas ao nível de infraestrutura, exigindo, portanto, a intervenção constante de operadores humanos.

O trabalho de Cao et al. (2018) apresenta uma solução para automatizar a migração de VMs no âmbito de ambientes *OpenStack* ([www.openstack.org](http://www.openstack.org)). Os autores propuseram um modelo que permite inferir os hábitos de consumo durante migrações, visando reduzir o tempo de interrupção de serviços (*downtime*) e também a percepção de queda de desempenho do CS (*slowdown*). Para isso, é construído um Modelo de Predição de Usuário (*User Behavior Prediction* - UBP). A predição do comportamento dos usuários durante uma migração é utilizada tanto para a escolha de uma VM migrante como também de um *host* de destino.

A coleta de dados ao nível da infraestrutura foi executada por uma API fornecida por terceiros (*Collectd* - <https://collectd.org/>) que permite a coleta periódica de métricas tanto do Sistema Operacional como das aplicações em execução nas VMs. Tais dados são inicialmente usados para localizar violações de *thresholds*. De fato, essa é a única estratégia adotada no trabalho de Cao et al. (2018) para a identificação e verificação de condições de migração. A escolha de VMs migrantes é baseada na predição dos valores do uso de cada recurso. Esses valores são dispostos em uma matriz de pesos que permite ordenar as VMs conforme o peso do uso de cada recurso.

A escolha de um destino também baseia-se nas predições de uso de recursos tanto pela VM ( $M_1$ ) como do *host* ( $M_p$ ). Ambas as predições não podem ultrapassar *thresholds* estabelecidos para cada recurso ( $M_{th}$ ), ou seja, a migração não pode afetar os CSs que já estejam em curso nas VMs do destino. Dessa forma, tem-se que  $M_1 + M_p < M_{th}$ .

Outro trabalho que se baseia no aprendizado do comportamento é apresentado em (Basu et al., 2019). De fato, um paradigma da aprendizagem de máquina é utilizado, o Reforço de Aprendizado (*Reinforcement Learning* - RL). A ideia é permitir que o comportamento dinâmico de *workloads* seja aprendido, permitindo que decisões de migração diminuam o potencial do aumento do consumo de energia e também das violações de SLAs.

Para (Basu et al., 2019), a previsão de carga para uma VM migrante assim como para o *host* selecionado como novo destino era, em geral, baseada em dados estáticos, desconsiderando, portanto, a previsão de variação do comportamento no uso de recursos. Tais limitações motivaram a adoção de RL como estratégia para aprendizado do comportamento dos *workloads*. O algoritmo Megh foi idealizado para lidar com as limitações mencionadas. Processos de Decisão de Markov

são usados por um monitor RL, com base na compreensão do ambiente, permitindo que se escolha a VM migrante e o *host* destino adequados, cujo custo do consumo de energia e das violações de SLA sejam mínimos.

Migrações que sejam efetivas são aquelas que geram resultados duradouros. Ou seja, a instabilidade detectada em um determinado *host* é solucionada, de forma a não gerar novas necessidades de migração por causa do reajuste no mapeamento de recursos causado pela migração. Alguns trabalhos sugerem estratégias para que a escolha da VM migrante seja efetiva. Os algoritmos sugeridos por Beloglazov e Buyya (2012) devem ser executados inúmeras vezes até que o *host* analisado não apresente mais problemas. O trabalho de Cao et al. (2018) sugere que previsões sobre o uso de memória e dos acessos à VM durante a migração sejam usados para descrever uma matriz de pesos. A VM com o menor peso é selecionada para migrar, ou seja, prioriza-se migrar as VMs que têm menor propensão de gerar páginas de memória sujas. Da mesma forma, (Basu et al., 2019) também apresenta uma estratégia para que a migração produza resultados efetivos.

O atraso, por menor que seja, entre a ocorrência da condição de migração e sua detecção assim como a queda de desempenho durante a migração são fatos cujas consequências são dificilmente corrigidas. Por esse motivo, Basu et al. (2019) sugerem que a previsão do comportamento dos *workloads* seja utilizado de forma a contribuir para migrações mais efetivas, ou seja, cujos resultados tendem ser mais duradouros.

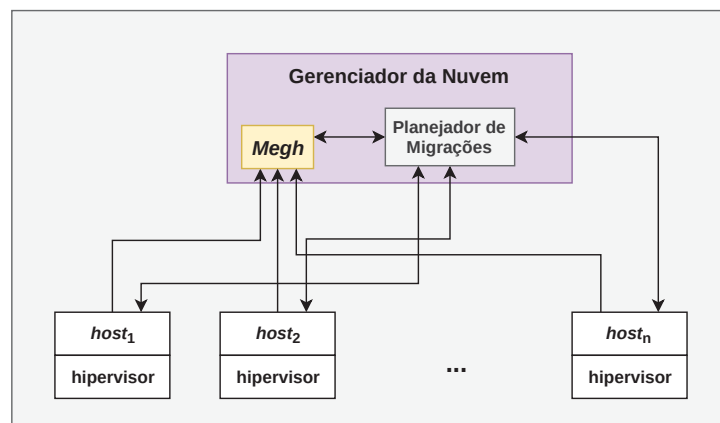


Figura 4.5: Planejamento de migrações com o suporte de *Megh*.

*Megh* é implementada como parte de um gerenciador de nuvem. A ideia é que diferentes hipervisores, ligados aos *hosts* do domínio, comuniquem periodicamente informações sobre a dinâmica dos *workloads*, conforme descrito na Figura 4.5. O gerenciador de nuvem recebe tais informações e o algoritmo *Megh* aprende sobre essa dinâmica. O aprendizado é utilizado por algum componente que constrói planos de migração, cujo foco é a redução do consumo de energia e das violações de SLA. Tais planos são conduzidos aos respectivos *hosts*. Em comparação ao MMT, os resultados demonstram que, através de *Megh*, o aprendizado do ambiente realmente diminui o custo da migração assim como a quantidade de migrações que ocorrem dentro de um intervalo de tempo. Além disso, as contribuições da abordagem apresentada por Basu et al. (2019) permitem atestar a importância de decisões de migração que garantam sua efetividade. Contudo, o aprendizado do comportamento dos *workloads* é processado por uma entidade central, que gerencia a nuvem. Essa estratégia tende a sobrecarregar o gerenciador da nuvem e diminuir as contribuições esperadas.

### 4.3 GERÊNCIA DE MIGRAÇÕES INTER-DC

Nesta seção são apresentados os esforços para que se estabeleçam soluções para o gerenciamento dinâmico de recursos, baseado na migração de VMs, entre diferentes DCs. É importante salientar que dois cenários podem ser identificados: DCs na mesma nuvem e DCs de diferentes nuvens (inter-nuvem). Em ambos os casos, a migração de VMs é confrontada com desafios técnicos. Por exemplo, na migração de VMs em redes de longa distância, obstáculos como: a manutenção do endereçamento de rede, a transferência de massivas quantidades de memória e a heterogeneidade dos distintos ambientes por onde a VM migrante deve passar, diminuem o potencial de que sejam atendidos os objetivos no gerenciamento dinâmico de recursos.

#### 4.3.1 Reservoir

Do nosso conhecimento, *Reservoir* (Rochwerger et al., 2009a) é a primeira abordagem para o estabelecimento de uma Federação de Nuvens Computacionais, com o propósito de aproximar CPs que atuam ao nível de IaaS.

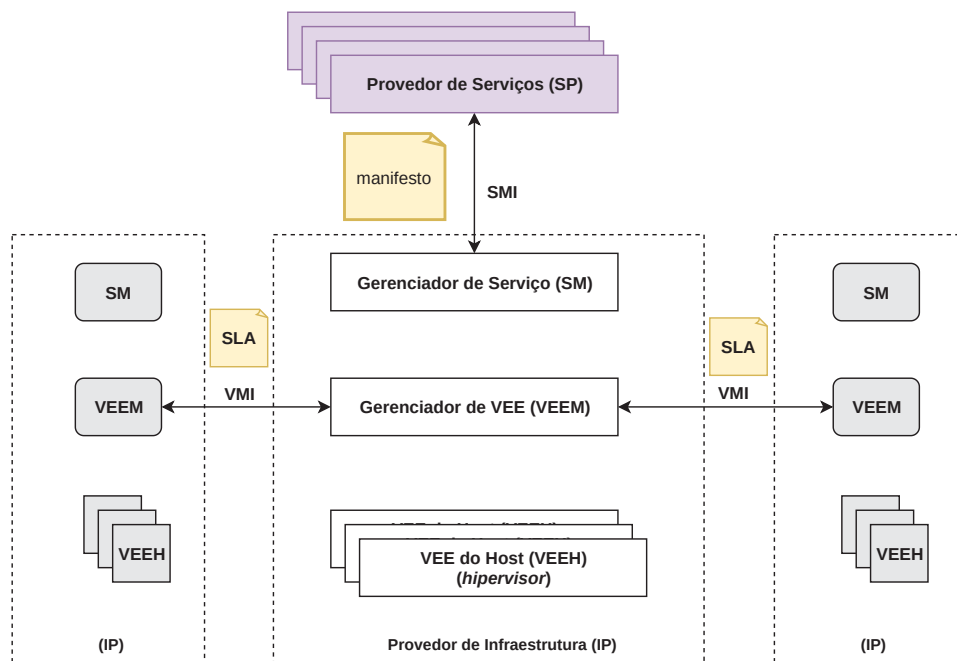


Figura 4.6: Arquitetura do *framework* Reservoir (Rochwerger et al., 2009b)

A Figura 4.6 apresenta os principais componentes da arquitetura de *Reservoir*. Um *Service Provider* é o correspondente ao CSP e sua interação com o *framework* se dá com a entidade *Infrastructure Provider* (que é o correspondente ao CP), através de um componente designado como *Service Manager* (SM). É usual que o Provedor de Infraestrutura não consiga assumir a capacidade de recursos requerida no Manifesto e, por esse motivo, precisa utilizar das Interfaces Virtuais de Gerenciamento (VMIs) para encontrar outros membros que possam atender ao pedido do provedor de serviço.

No âmbito de cada domínio *Reservoir*, VMs são designadas como *Virtual Environment Entities* (VEE), sendo coordenadas pelo (*VEE Manager* - VEEM) que possui associação direta com o *hipervisor*. Dessa forma, um VEEM tem a capacidade de executar duas ações importantes: a alocação estática de VMs (consoante as restrições determinadas pelo SM) e a migração de VEEs entre os domínios da federação.

Para lidar com os desafios técnicos ao nível da camada de gerenciamento foi proposta a criação de um *Transfer Proxy* (TP) (Elmroth e Larsson, 2009). Cada *host* implementa um ou mais TPs, cujo objetivo é abstrair os dados de VMs, através de estruturas designadas como *Transfer Token* (TT). Cada TT mantém um mapa relacionando o identificador de uma VM ao seu hospedeiro, sendo utilizado na execução das migrações. Como dados sensíveis de uma VM não estão disponíveis em um TT, cada IP intermediário que participa de uma migração não tem conhecimento senão da origem da VM. Cada IP pode manter mais de um TP, permitindo que várias VMs do mesmo *host* possam migrar. Uma migração, ao final, envolve a transferência dos dados originais de uma VM para um TP distinto.

O trabalho de Rochwerger et al. (2009a) ressalta as dificuldades no gerenciamento dinâmico de recursos em federações de nuvens. As restrições técnicas impostas à migração de VMs entre DCs requer estratégias que diminuam a sobrecarga causada pela migração e que é potencializada pela necessidade de uma migração cruzar vários domínios. A solução apresentada foi diminuir a necessidade de inspeção recorrente de dados, mascarando dados sensíveis. No entanto, essa ação é vantajosa para a execução da migração, mas, a provisão de dados para o planejamento da migração não recebeu nenhuma alteração. Dessa forma, é razoável considerar que a solução teria bom desempenho na execução, mas, não no planejamento.

#### 4.3.2 InterCloud

*InterCloud* (Buyya et al., 2010) é uma iniciativa de federação de nuvens baseada na comercialização de recursos, disponíveis em diferentes nuvens. O propósito inicial de *InterCloud* é prover elasticidade a CSs por meio do auto-dimensionamento, ou seja, de uma expansão automática da capacidade dos recursos das VMs associadas ao CS.

A predição de comportamento é importante para que decisões de auto-dimensionamento não sejam reativas, contribuindo para que o cumprimento de SLAs não seja comprometido. O mapeamento de recursos tende a variar, demandando soluções que lidem com alterações constantes, mas, que não prejudiquem o desempenho. Por essa razão, Otimização Multi-Objetivo (*Multi-Objective Optimization* - MOO) é empregada, buscando lidar com objetivos de otimização focados no uso de recursos, tais como disponibilidade e confiabilidade, mas que também atendam aos interesses do usuário, tais como tempo de resposta e custo econômico. No tocante à interoperabilidade, (Calheiros et al., 2011) destaca os diferentes padrões, especialmente, os dedicados à representação de imagens de VMs. Embora padrões tenham sido estabelecidos, tais como o *Open Virtualization Format* (OVF) (Jegou et al., 2012), ainda são necessários mecanismos para permitir lidar com a interoperabilidade de VMs em domínios como uma federação de nuvens.

Na Figura 4.7 é apresentada a arquitetura de *InterCloud*, contendo os seguintes componentes: *Cloud Coordinator* (CC), *Cloud Broker* (CB) e *Cloud Exchange* (CEX). É a partir de um CC que intermediações ocorrem, tanto entre consumidor e CP, assim como entre CPs e os respectivos CSPs.

CBs são entidades responsáveis por dirimir a comunicação entre consumidor e a *InterCloud*. É conveniente ressaltar que um consumidor é visto não apenas como um consumidor de CSs, mas, também como um CSP. Por esse motivo, CBs podem assumir duas perspectivas e que estão associadas a diferentes formatos de comunicação. O CEX atua como um orquestrador do mercado de recursos, provendo informações sobre disponibilidade e capacidade de recursos. Ou seja, um CEX mantém informações detalhadas dos recursos e provê aos CCs a capacidade de decisão sobre novas alocações de recursos.

CEX é um componente centralizador na arquitetura, responsável por manter as ofertas de recursos em bancos de dados alimentados por CCs. Quaisquer requisições de recursos efetuadas

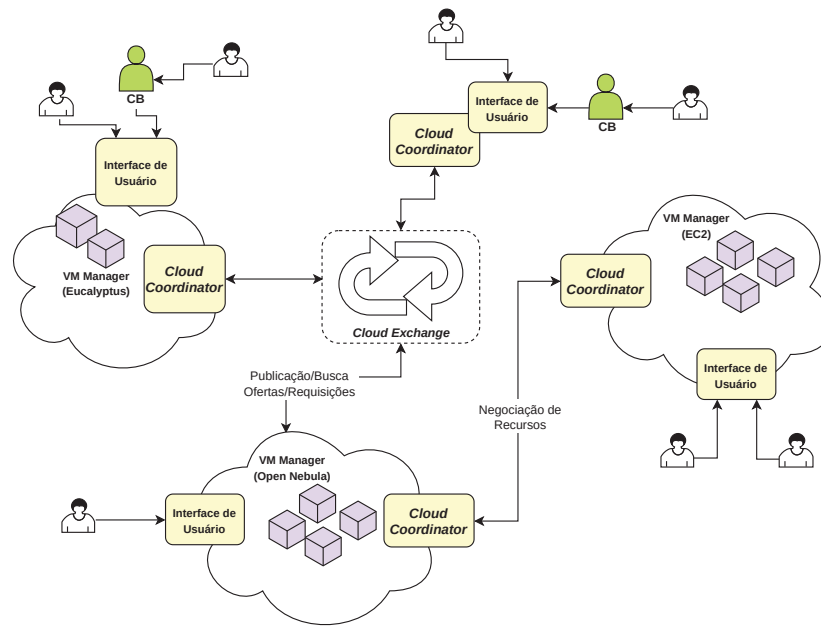


Figura 4.7: Arquitetura do *framework* InterCloud. Adaptado de (Calheiros et al., 2012).

por consumidores, utilizam de CBs para acesso aos CCs que, por sua vez, comunicam-se entre si na busca por VMs que atendam às requisições. Há, entretanto, outra importante ação praticada pelos CCs, a comunicação ao CEx da disponibilidade de um recurso.

Cada CC é responsável pelo gerenciamento de um ambiente de nuvem, dispondo de interfaces para comunicação com o CEx e com outros CCs, como descrito na Figura 4.8. Cada CC acessa um Gerenciador de VMs (*VM Manager* (VMMg)) para decisões relacionadas ao gerenciamento de recursos. As VMs de cada nuvem podem ser gerenciadas por diferentes VMMgs, tais como *Amazon EC2*, *Eucalyptus* (<http://eucalyptus.com>), *OpenNebula* (<https://opennnebula.org/>) e *Aneka* (<http://www.manjrasoft.com/products.html>).

A alocação de VMs assim como o escalonamento nos diferentes *hosts* depende de ações do CC. Como parte das ações de gerenciamento do CC, sensores são utilizados para averiguar dois objetivos de gerenciamento: economia de energia e minimização das violações de SLA. Em caso de violações de *thresholds*, o CC dispara para o CEx uma consulta de disponibilidade de recursos que pode requerer o disparo de uma migração como também de um auto-dimensionamento.

Dentre as funcionalidades associadas à virtualização, o CC implementa um Gerenciador de Mobilidade (*Mobility Manager* - MM) (Buyya et al., 2010). MM analisa periodicamente dados fornecidos pelos sensores, estando apto a gerar planos de migração. No entanto, limitações técnicas impostas por hipervisores, naquele momento, contribuíram para que as decisões de migração fossem restritas a um domínio intra-DC. De fato, como ressaltada por Buyya et al. (2010), hipervisores como Xen (<https://xenproject.org/>) e VMWare (<https://www.vmware.com/>), à época, apenas executavam migrações na mesma subrede e com o apoio de um compartilhamento de armazenamento.

Uma alternativa para a provisão dinâmica de recursos é o auto-dimensionamento. Essa alternativa é adotada no trabalho de Calheiros et al. (2011). CCs implementam entidades que podem ser criadas dinamicamente, designadas Clientes CC (Figura 4.8). Essas entidades possibilitam que haja interação com outros CCs e também a comercialização de novos recursos.

O impacto do auto-dimensionamento como resposta para a elasticidade dos recursos em nuvens é verificado por Calheiros et al. (2011), sendo considerado em um cenário de pequena escala, uma solução de bom desempenho. Contudo, é importante salientar que a escalabilidade



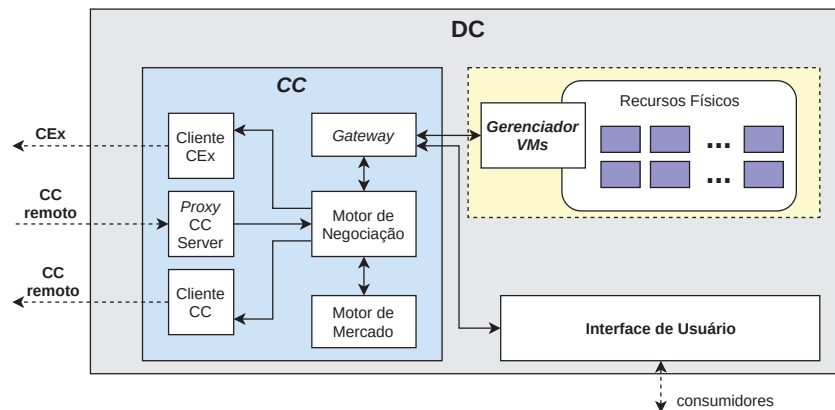


Figura 4.8: Arquitetura do componente *Cloud Coordinator*. Adaptado de Calheiros et al. (2012).

da solução não foi avaliada. Além disso, deve-se ressaltar a complexidade do espalhamento de um recurso auto-dimensionado. Como diferentes domínios estarão envolvidos, monitorar e controlar o uso de tal recurso torna-se uma tarefa mais complexa ainda à medida que esse cenário cresce.

#### 4.3.3 xVMotion

*XvMotion* (Mashtizadeh et al., 2014) é um sistema que permite que migrações ocorram entre pontos que não estejam na mesma rede, portanto, podendo ser utilizado em migrações inter-DC. Como descrito na Figura 4.9, distintos domínios são interconectados por entidades designadas *Streams* (St). Cada St age como um protocolo de transporte baseado no TCP e cujo propósito é transferir páginas de memória e conteúdo de armazenamento de VMs entre diferentes *hosts*. Como mais de uma migração pode ocorrer, Sts podem lidar com várias conexões.

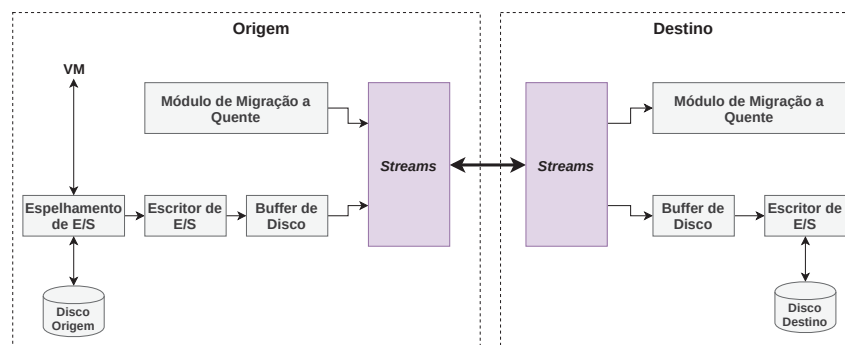


Figura 4.9: Arquitetura da solução de migração a longa distância *XvMotion*. Adaptado de (Mashtizadeh et al., 2014).

As páginas de memória que devem ser transferidas são localizadas pelo Módulo de Migração a Quente (MMQ) que também controla a fila de páginas de memória que devem ser enviadas ao St. O conteúdo do disco virtual usado pela VM migrante é espelhado no destino, no entanto, para impedir que as páginas de memória que são modificadas (ou seja, páginas sujas de memória) tenham conteúdo desatualizado transferido durante a migração, o componente Espelhamento de E/S intercepta cada escrita no disco virtual e promove um espelhamento assíncrono com o disco virtual do destino. O conteúdo é, então, destinado para o Escritor de E/S que prepara o conteúdo a ser transferido no *Buffer de Disco*.

Três premissas são atendidas pela estratégia de Mashtizadeh et al. (2014): o efeito do tempo de interrupção do serviço associado à VM (*downtime*) não pode prejudicar o consumidor; a

troca de contexto, ou seja, a mudança do *workload* em execução da VM tem que ser imperceptível para o consumidor e a migração não deve requerer a interferência de provedores intermediários, ou seja, a comunicação entre a origem e o destino é ponto-a-ponto.

Conforme (Mashtizadeh et al., 2014), os resultados de *XvMotion* constataam seu bom desempenho mesmo em *links* com alta latência. Além disso, foi constatado que a transferência em conjunto de páginas de memória e espaços de armazenamento apresentaram baixo *downtime* dos *workloads* em execução nas VMs migrantes. Ou seja, é razoável considerar que *XvMotion* é uma alternativa que poderia impulsionar a adoção da migração de VMs inter-DC em um *framework* como o *InterCloud*.

#### 4.3.4 ACTiCloud

Pode-se considerar que a relação entre consumo de recursos e CSs sejam de dois tipos: CSs que se adequam ao conjunto de recursos alocados nas VMs associadas e CSs cuja demanda de recursos expande de maneira desordenada, ou seja, cujos *workloads* flutuam de forma imprevisível. Para a segunda categoria é comum que os recursos fltuem entre a sobrecarga e a ociosidade com frequência, exigindo estratégias para a readequação da oferta de recursos que, em geral, se dá por meio da migração de VMs.

O hardware tradicional de computadores, baseado na interconexão dos distintos recursos por meio de uma placa-mãe é considerada por Goumas et al. (2017) um limitador para o segundo tipo de CSs. Como a alocação dos recursos fica delimitada aos *hosts*, é razoável considerar que a disponibilidade de recursos tende a demorar para ser percebida. Ou seja, situações como a fragmentação e a subutilização de recursos podem decorrer dessa organização.

Uma alternativa para lidar com as consequências indesejadas dos CSs que fltuam desordenadamente é a estratégia conhecida como desagregação de recursos (Han et al., 2013). Ou seja, lidar com a estruturação de DCs de forma que ao invés dos recursos serem confinados em *hosts*, serem tratados com entidades isoladas e, portanto, com maior autonomia e também oferecendo maior flexibilidade para as soluções de gerenciamento de recursos. Essa visão é a base do *framework ACTiCloud* (Goumas et al., 2018).

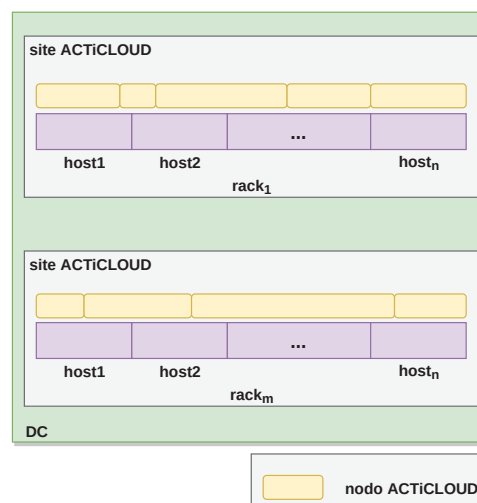


Figura 4.10: Nodos (grupos de recursos) em um DC *ACTiCloud*. Adaptado de (Goumas et al., 2017).

Conforme descrito na Figura 4.11, a base da arquitetura *ACTiCloud* é uma camada designada Recursos de Hardware Desagregados (RWD) que possui mecanismos para construir conjuntos de recursos ao nível dos *racks* dos DCs. Como descrito na Figura 4.10, nodos são os



grupos de recursos resultantes da desagregação, comportando VMs e sendo, portanto, a entidade atômica gerenciada por hipervisores na arquitetura *ACTiCloud*. *Sites* identificam o agrupamento de diferentes nodos, sendo os destinos para a alocação e também para a migração de VMs.

A camada de Agregação usa da organização provida pela camada RWD para disponibilizar recursos em conjuntos gerenciáveis à camada de Hipervisor que possibilita a virtualização dos recursos agregados e os oferece como VMs ao Gerenciamento Holístico de Recursos (GHR).

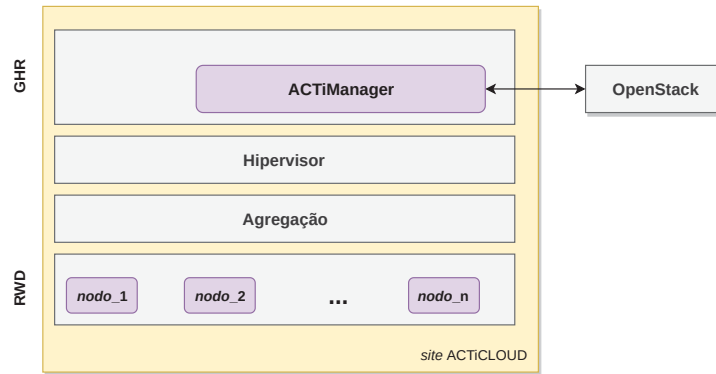


Figura 4.11: Arquitetura de um *host* baseado em *ACTiCLOUD*. Adaptado de (Goumas et al., 2018).

*ACTiCLOUD* apresenta uma estratégia de gerência dinâmica de recursos integrada a *OpenStack*. A entidade designada *ACTiManager*, descrita na Figura 4.12, é a responsável por executar, monitorar, migrar e encerrar CSs. O monitoramento dos CSs envolve acompanhar os *workloads* em execução, por isso, é também tarefa do *ACTiManager* o controle dos grupos de recursos.

Para Goumas et al. (2018), os trabalhos anteriores que lidam com o gerenciamento dinâmico de recursos em nuvens tendem a tratar a alocação de *workloads* da mesma maneira. Em geral, busca-se consolidar *workloads* em VMs de forma a se utilizar o mínimo de *hosts*. No entanto, essa estratégia estimula que *workloads* dependentes do uso intenso de recursos necessitem estar co-localizados no mesmo *host*, gerando necessidades constantes de migração.

Trabalhos de pesquisa anteriores já propuseram esquemas que buscam otimizar a coexistência de *workloads* e, ao mesmo tempo, tentam impedir a interferência entre as VMs por meio da elaboração de perfis do uso dos recursos. No entanto, Goumas et al. (2017) ressaltam que Sistemas Operacionais de Nuvem ainda não lidam com essa questão de forma automática, delegando a administradores do ambiente a coleta e a verificação das condições de uso dos recursos pelas VMs. Esse é o caso de *OpenStack* e, por esse motivo, *ACTiCloud* visa estender suas tarefas de gerenciamento de recursos e automatizar decisões como a migração de VMs.

Sempre que uma VM é criada, ela é mantida em um estado designado "em laboratório". A ideia é que *ACTiManager* possa coletar informações sobre a VM para depois inferir sobre seu comportamento padrão, se ela tende a usar parte de seus recursos de forma massiva e, portanto, podendo afetar outras VMs e também se sua execução pode ser interrompida. Ou se o seu desempenho é rapidamente afetado por alterações no uso dos recursos alocados. Sempre que a etapa de "laboratório" é encerrada, o perfil inferido de uma VM pode influenciar na execução de migrações. Contudo, outras condições de migração são consideradas: se há um cenário de interferência de VMs e se ocorre subutilização de recursos. Uma interferência é caracterizada pela perda de desempenho de outras VMs, co-localizadas no mesmo *site*, em decorrência do uso de um ou mais recursos acima de um determinado *threshold*. A subutilização é identificada da mesma forma, através de *thresholds* que indicam recursos sendo desperdiçados.

Para se identificar condições de migração, *ACTiManager* é invocado em períodos de tempo específicos, definidos por um operador humano, possibilitando que o comportamento de

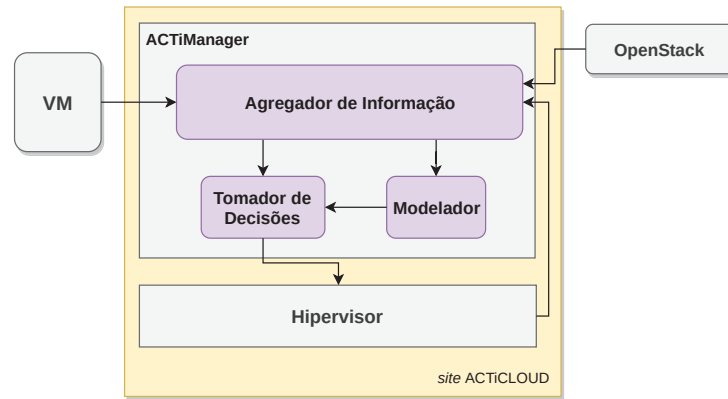


Figura 4.12: Arquitetura do *ACTiManager*. Adaptado de (Goumas et al., 2018).

cada VM durante esse período possa ser inferido. Assim, decisões de migração não são baseadas em comportamentos pontuais, indicando uma tendência de anomalia. Além disso, técnicas de predição e algoritmos para ordenação das migrações são mencionados no documento de Goumas et al. (2018). Elas são geradas pelo Modelador, descrito na Figura 4.12 e usadas pelo Tomador de Decisões que implementa algoritmos para o disparo de migrações e também para decidir a ordem de migração com base nas predições.

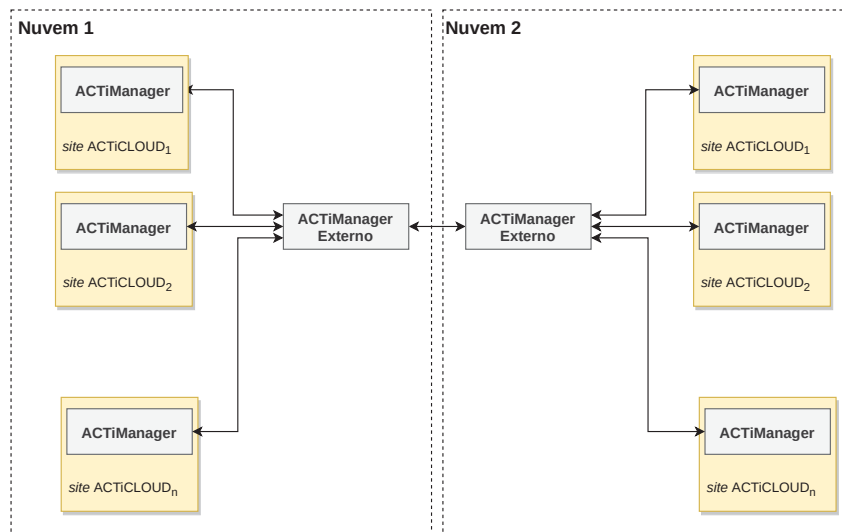


Figura 4.13: Arquitetura do ambiente de gerenciamento de recursos inter-DC no *ACTiManager*. Adaptado de (Goumas et al., 2018).

Os diferentes *ACTiManager* em cada *site* integram-se por meio de um *ACTiManager Externo*, entidade de gerenciamento de cada nuvem, conforme descrito na Figura 4.13. Ambos gerenciadores, seja ao nível de *site* como o designado Externo possuem a mesma arquitetura de componentes. A diferença está na interação com *OpenStack*. *ACTiManagers* internos apenas podem coletar dados de monitoramento, enquanto os externos podem requisitar informações.

O *ACTiManager Externo* pode ser visto como um módulo que estende *OpenStack*, usando de suas APIs e interfaces tanto para processar dados do comportamento de VMs como também para informar e requerer ações de remapeamento de recursos. Contudo, assim como *OpenStack*, *ACTiCLOUD* também encontra barreiras para lidar com migrações entre DCs de domínios geográficos e administrativos distintos. No entanto, recentemente, no trabalho de

Kollberg et al. (2020) é apresentada uma abordagem para migração entre distintos DCs, baseados na proposta *ACTiCLOUD*.

#### 4.3.5 HeatSpreader

O trabalho de Kollberg et al. (2020) apresenta uma abordagem para a integração de múltiplas nuvens, baseada na infraestrutura fornecida pelo Sistema Operacional de Nuvens *OpenStack*. Serviços como *Heat* (<https://docs.openstack.org/heat/latest/>) que promove a orquestração de VMs e *Nova* (<https://docs.openstack.org/nova/latest/>) que possibilita a criação e manutenção de VMs, são estendidos para permitir que colaborações contribuem na execução de operações de manutenção e solução de falhas.

Conforme descrito na Figura 4.14, um Controlador Multi-Nuvem (CMN) foi desenvolvido, buscando lidar com uma quantidade pré-definida de nuvens, estabelecida por um operador. Cada *host* utiliza do Orquestrador de Nuvens *OpenStack-Heat* para o controle de CSs, especificamente dos recursos que estão associados a eles. Por isso, a Pilha *Heat* (PH), descrita na Figura 4.14 diz respeito ao domínio de controle e de onde são identificadas situações adversas tais como a sobrecarga de um *host*. Tais situações são identificadas pelo Agente de Sobrecarga (AgS) que comunica ao Controlador de Sobrecarga (CtS) do *host*.

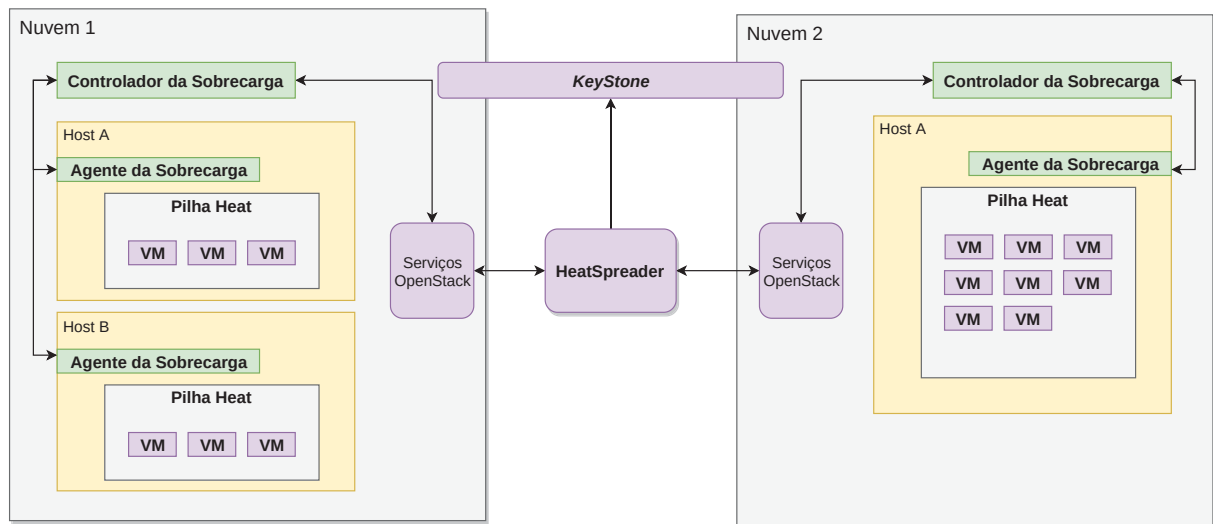


Figura 4.14: Arquitetura do ambiente de gerenciamento de recursos inter-DC no *HeatSpreader*. Adaptado de (Kollberg et al., 2020).

*Thresholds* são usados pelos AgSs para identificar sobrecargas, requerendo decisões do CtS. No âmbito do CMN, os CtSs podem decidir por migrações, no entanto, o objetivo principal não é a execução de uma migração isolada, mas, a da transferência de várias VMs de forma a manter a carga de um *site* dentro de um limite (*threshold*) definido pelo operador. Contudo, o balanceamento de carga requerido por um CtS pode não ser possível no âmbito de um *site*, por isso, nesse caso, o *HeatSpreader* é contactado.

O componente central da solução (*HeatSpreader*) gerencia os diferentes CtS. Ou seja, enquanto CtS gerenciam uma nuvem, *HeatSpreader* é responsável por redistribuir VMs entre múltiplos *sites*, em nuvens diferentes. *HeatSpreader* não mantém atualizações do consumo de recursos de cada *site*, essa tarefa é dedicada a cada CtS, mas, disponibiliza interfaces que podem ser usadas externamente para que cada CtS informem esse *status*. Isso ocorre, periodicamente, em intervalos de tempo definidos pelo operador.

Uma limitação imposta por *HeatSpreader* é que migrações inter-DC só podem ocorrer com VMs sem um estado associado. Ou seja, apenas a imagem da VM é transferida. De fato, como mencionado anteriormente, o custo computacional e a complexidade de uma migração inter-DC têm sido dificultadores. Apesar de estratégias terem sido propostas, é razoável considerar que elas dependem de contextos específicos, exigem adequações técnicas, além de terem um custo computacional e escalabilidade muitas vezes desconhecidos.

A concepção de uma federação proposta no trabalho de Kollberg et al. (2020) requer que três requisitos sejam atendidos: um operador humano precisa definir antecipadamente que CPs participam; todos devem ser baseados em *OpenStack* e *HeatSpreader* precisa ser executado externamente a todas as nuvens parceiras. Em comparação a outras iniciativas de estabelecimento de nuvens federadas, tais como: (Rochwerger et al., 2009b), (Buyya et al., 2010), (Carlini et al., 2011), (Bernstein e Demchenko, 2013), etc., o trabalho de Kollberg et al. (2020) pode ser considerada uma solução leve por não requerer adaptações e nem camadas extras para tradução de instruções.

#### 4.4 DESAFIOS NA GERÊNCIA DA MIGRAÇÃO DE VMS

Os trabalhos apresentados anteriormente destacam ações que visam contribuir tanto com o planejamento como com o controle e a execução das migrações. Pode-se perceber que algumas ações são adotadas em ambas as etapas da migração de uma VM. No entanto, alguns desafios que dificultam a implementação destas ações são aqui descritos e posteriormente discutidos.

- (A1): Monitoramento com critérios para disparar migração - mecanismo que permite certificar que um alerta de necessidade de migração não advém de um falso positivo, como por exemplo, um pico momentâneo no uso de recursos que poderia demandar o início do planejamento de uma migração;
- (A2): Minimização da interferência de um operador humano - ação que contribui para a eliminação da necessidade de interferência de um operador humano;
- (A3): Uso de mais de um *threshold* - adoção de diferentes valores (estáticos ou dinâmicos) para o controle da identificação de *hosts* instáveis seja para iniciar como executar o planejamento de uma migração;
- (A4): Controle distribuído - mecanismos utilizados para que a decisão de migrar ou executar uma migração não sejam confinados a uma entidade centralizada;
- (A5): Controle da execução dos planos de migração - alguma estratégia que contribui para que os dados associados à execução de uma migração possam ser utilizados na produção de melhores planos de migração e também que permitam que, ao ser executada, uma migração não incorra em prejuízos ao ambiente;
- (A6): Planejamento das migrações efetivas - ação que possibilita a criação de planos de migração que tornam o efeito de reequilibrar os recursos mais duradouro.

Todos os trabalhos sugerem alguma estratégia para que a coleta gere dados que possam ser considerados confiáveis, ou seja, que não estejam desatualizados e que descrevam o real panorama no uso dos recursos. Para atingir esse propósito, duas estratégias são identificadas: janelas deslizantes de tempo (*W*) e intervalos de tempo dedicados à análise do perfil da VM. Em ambos os casos, análises estatísticas são utilizadas para confirmar tendências de comportamento e também para prever o uso de recursos após a migração.

Na maioria dos trabalhos a duração de uma janela  $W$  é definida como um valor constante. No entanto, Xu e Fortes (2011) chamam atenção para os potenciais prejuízos dessa abordagem, ressaltando que o comportamento variável e imprevisível dos *workloads* não era considerado com esta abordagem. Para tanto, sugeriu a definição de Intervalo Estável ( $T_{stable}$ ), um valor que pode ser aproximado com base na duração esperada de uma migração. Conforme (Xu e Fortes, 2011)  $T_{stable} = W > T_{migration}$ , ou seja, o uso dos recursos por uma VM e por um *host* mantêm-se estáveis por um período de tempo que é superior ao tempo gasto com uma migração. A discussão levantada por Xu e Fortes (2011) é relevante, no entanto, a solução apresentada é limitada, pois a aproximação do valor de  $T_{stable}$  é aleatório, baseado em um tamanho de VM (quantidade de uso de memória) que não, necessariamente, representa o real status.

Trabalhos que focam na automação da migração de VMs, em geral, estabelecem mecanismos que permitem gerar planos automáticos de migração, diminuindo a necessidade de um operador coletar dados para análise de condições de migração e nem avaliar as potenciais opções para os planos de migração. Por outro lado, pode ser observado que na execução da migração, há uma lacuna no tocante ao controle da migração. Essa lacuna impacta na análise das condições de migração e também pode contribuir para fenômenos indesejados, tais como o aumento do consumo de energia (Baccarelli et al., 2015).

Tabela 4.1: Contribuições para a elaboração e execução de migrações

Trabalho	A1	A2	A3	A4	A5	A6
(Wood et al., 2007)	x	x				x
(Xu e Fortes, 2011)	x	x	x			x
(Beloglazov e Buyya, 2012)	x	x	x			x
(Baccarelli et al., 2015)	x	x	x		x	x
(Cao et al., 2018)	x	x		x	x	
(Basu et al., 2019)					x	x
(Rochwerger et al., 2009b)		x	x			
(Buyya et al., 2010)	x	x	x	x		
(Mashtizadeh et al., 2014)		x			x	x
(Goumas et al., 2017)	x	x	x	x		x
(Kollberg et al., 2020)	x	x	x	x		x

*Thresholds* são úteis na detecção de eventos. No planejamento de migrações, eles são utilizados para identificar condições de migração. Contudo, o inevitável comportamento variável dos *workloads* pode requerer duas ações em relação aos *thresholds*. A primeira é determinar um *threshold* que averigua sobrecargas e outro para analisar subutilizações de recursos. Enquanto o primeiro pode ser utilizado para impedir a interrupção de CSs por falta de recursos nas suas VMs associadas, o outro *threshold* permite atingir objetivos de gerenciamento, tais como economia de energia. Contudo, *thresholds* pode ser dinâmicos, tais como nos trabalhos de (Xu e Fortes, 2011; Beloglazov e Buyya, 2012; Cao et al., 2018). Neste caso, são adequados para espelhar o dinamismo dos *workloads*, sendo necessário capturar a tendência de comportamento do ambiente.

No trabalho de Beloglazov e Buyya (2012) são consideradas revisões periódicas dos valores de *thresholds*. O Desvio Médio Absoluto (DMA) do uso de CPU é gerado e utilizado para calibrar o *threshold* usado para averiguar sobrecargas. Quanto maior DMA, menor é o *threshold*. De fato, à medida que aumenta o DMA, maior é a probabilidade de que alguma violação de SLA ocorra. As mudanças de valores de *threshold* precisam ser comunicadas aos *hosts* do DC, no entanto, não se observa nenhuma estratégia para garantia da confiabilidade da informação divulgada.

As consequências de não se conduzir migrações sob um controle distribuído também são ressaltadas. Diferentes autores citados apresentam uma abordagem distribuída de controle de migrações. No trabalho de Zhang et al. (2018a), migrações são planejadas e apenas origem e destino participam de negociações para a transferência dos dados. Essa estratégia no entanto, é conveniente quando não é necessário migrar várias VMs, sendo, portanto, uma solução não escalável. Nos trabalhos de Buyya et al. (2010); Kollberg et al. (2020), são apresentados esquemas para a distribuição do controle. No entanto, ambos requerem que acordos prévios tenham sido estabelecidos a fim de que parcerias entre CPs pudessem ser firmadas. De fato, até onde pode ser observado, ainda não foi apresentada uma solução que lide com parcerias instantâneas entre CPs e que, portanto, não exigem uma infraestrutura padrão para participação em federações de nuvens. Além disso, é importante ressaltar que ambas as soluções mencionadas adotam um controle hierárquico e distribuído.

Outro desafio é lidar com a execução de um plano de migração que, geralmente, tem o potencial de gerar custos indesejados, tais como o aumento do consumo de energia, a geração de calor e violações de SLAs. Em geral, tais problemas somente são percebidos durante a execução do plano. Alguns trabalhos provêm soluções para esse cenário, visando reduzir os impactos durante a migração (Baccarelli et al., 2015). Em outros trabalhos, o impacto da execução de uma migração é antecipado e incorporado no planejamento da migração. Por exemplo, nos trabalhos de Cao et al. (2018); Zhang et al. (2018a); Basu et al. (2019), dados relativos ao uso dos recursos são monitorados e controlados durante a migração, mas, no intuito de prover modelos de predição que contribuam na escolha de VMs migrantes e de *hosts* de destino.

A solução apresentada por Baccarelli et al. (2015) permite diminuir o consumo de energia pelo gerenciamento da largura de banda durante uma migração. As outras soluções apresentadas usam das informações da execução de planos de migração visando diminuir potenciais *slowdown* e *downtime* na provisão de CSs. Uma preocupação semelhante também constatada no trabalho de Wood et al. (2007), cuja solução, no entanto, é baseada em modelos estatísticos de predição que permitiam antecipar a necessidade de recursos de VMs, permitindo incluir essa informação na escolha de VMs migrantes.

Informações faltantes ou que não representam o estado corrente dos recursos contribuem para a produção de planos de migração que não solucionam a instabilidade de oferta de recursos. Por esse motivo, é necessário considerar estratégias que diminuam o potencial de uma migração não ser efetiva. Para Wood et al. (2007), a predição do comportamento de *hosts* contribui na escolha de VMs migrantes. Da mesma forma, Beloglazov e Buyya (2012) estabelece políticas baseadas na mesma perspectiva e ainda sugere que planos de migração sejam construídas enquanto não houver uma constatação de que um *host* tenha seu cenário, a curto prazo, definitivamente reequilibrado. Para Baccarelli et al. (2015), o controle da largura de banda utilizada pela migração não só diminui o potencial de maior consumo de energia, como também possibilita o acompanhamento dos dados dos *hosts* durante a execução de uma migração, permitindo que planos de migração não sejam baseados apenas em dados estáticos. Para Basu et al. (2019), a dinâmica dos *workloads* precisa ser capturada e por isso, seu agente de aprendizado *Megh* tem a capacidade de inferir como um *workload* se comportará e, portanto, tornar os planos de migração mais efetivos. A perspectiva de (Basu et al., 2019) pode ser identificada no trabalho de (Wood et al., 2007). Enquanto o primeiro lida com a dinâmica ao nível dos *workloads*, o outro lida ao nível de *hosts*.



## 4.5 CONSIDERAÇÕES FINAIS

Diferentes ações são recorrentes em trabalhos que focam em aspectos associados às migrações de VMs com vistas ao gerenciamento dinâmico de recursos. No entanto, alguns desafios ainda permanecem sem solução ou baseados em soluções restritivas.

A migração de VMs entre DCs é um dos desafios que se mantém em aberto. Restrições técnicas dificultam a proposição de soluções abrangentes, exigindo que outras restrições sejam consideradas. Algumas alternativas para contornar essa dificuldade envolvem adaptações à infraestrutura de DCs e VMs que, entretanto, não demonstram escalabilidade apesar do bom desempenho. Outras, exigem restrições que promovem uma rígida seleção dos potenciais ambientes que podem participar da solução, excluindo opções que poderiam solucionar o problema.

Dentre os trabalhos que usam as informações geradas durante a execução de planos de migração, três trabalhos mencionam que tais dados contribuem na redução de *slowdown* e *downtime*. No entanto, tais trabalhos são baseados em adequações da infraestrutura dos DCs e na utilização de técnicas específicas associadas às adequações. Outro trabalho, destaca-se por gerenciar a execução dos planos de migração, incorporando um gerenciador de largura de banda que controla a execução das migrações. Esta estratégia, no entanto, requer alterações significativas na condução da migração, requerendo adaptações à forma como cada página de memória é transferida. Dada a fina granularidade exigida pelas adequações, é razoável considerar que essa não seria uma solução escalável no controle de muitas migrações simultâneas. Ambas as limitações, observadas do uso das informações decorrentes da execução de planos de migração, parecem estabelecer obstáculos ao uso de tal conjunto de informações em soluções abrangentes.

Também é perceptível que as ações que visam contribuir no gerenciamento da migração de VMs têm sido implementadas de forma dispersa. Ou seja, contribuem que apenas parte das vantagens de uma migração possam ser sentidas na alocação dinâmica de recursos. Por exemplo, muitas soluções focam explicitamente na automação do planejamento da migração, negligenciando ações de gerenciamento que podem ser tomadas durante a execução.

É possível afirmar que as soluções anteriores se encaixem em ferramentas que solucionem questões como: a automação do planejamento da migração de VMs; a provisão de infraestrutura que possibilite uma migração intra e inter-DC e também para extrair melhores resultados com migrações sob infraestruturas de DC não convencionais. Em todas as categorias mencionadas é possível identificar esforços de gerenciamento tais como: monitoramento e controle dos recursos virtuais de cada VM e de cada *host*, decisões de migração baseadas em políticas focadas na eficácia da operação e a preocupação com o controle da informação coletada e utilizada no planejamento das migrações. Tais trabalhos estão, em geral, focados em apresentar ações de gerenciamento que contribuam para o planejamento das migrações, negligenciando a necessidade do controle da execução das migrações. É possível depreender que pouco tem sido feito para que as ações de gerenciamento integrem, de fato, informações relativas tanto ao planejamento como a execução da migração, contribuindo para planos de migração que tendem a não ser satisfatórios.

## 5 MODELO PARA A GERÊNCIA DA MIGRAÇÃO DE MÁQUINAS VIRTUAIS

Diferentes abordagens que apresentam ações para o gerenciamento da migração de VMs foram apresentadas no capítulo anterior. Como descrito, há uma lacuna na proposição de uma abordagem abrangente em que as ações de gerenciamento contemplem tanto as etapas do ciclo de vida de planejamento como da execução de uma migração. De fato, a ausência dessa integração contribui para que migrações sejam planejadas sob o suporte de informações equivocadas e faltantes.

Este capítulo apresenta um modelo para a gerência da migração de VMs, iniciando, na Seção 5.1 com uma descrição geral do modelo. Na Seção 5.2 são apresentadas as estratégias que esta tese considera para a detecção da necessidade de migrar VMs. A seleção das VMs que solucionam a necessidade de migração é apresentada na Seção 5.3. Para a conclusão do planejamento, na Seção 5.4 é apresentada a abordagem de seleção de *hosts* que permitem a elaboração do plano de migração.

A solução para o gerenciamento da migração de VMs apresentada nesta tese é baseada em um modelo descrito neste capítulo. Para que as definições do modelo proposto possam ser implementadas, no Capítulo 6 é apresentada a arquitetura para o gerenciamento da migração de VMs.

### 5.1 VISÃO GERAL DO MODELO

O gerenciamento da migração de VMs pode ser compreendido como a integração de diferentes ações que contribuem: (a) para que cada etapa do ciclo de vida de migração tenha à disposição dados e informações corretos e adequados; (b) para que o monitoramento e o controle ocorram tanto durante a elaboração do planejamento como em sua execução; (c) para que operadores humanos interfiram o mínimo nas ações de gerenciamento.

Migrações de VMs podem ocorrer em dois escopos: intra-DC ou inter-DC. O cenário intra-DC limita-se a um grupo de *hosts*, usualmente coordenados por um gerente de DC. A organização usual para a coordenação de migrações inter-DC envolve um gerenciador de nuvens, sendo a entidade responsável pela coordenação de migrações entre diferentes DCs. Existem dois cenários inter-DC: intra-nuvem e inter-nuvem. No primeiro cenário tem-se uma nuvem baseada em mais de um DC. No segundo cenário, os DCs estão espalhados por diferentes nuvens. Mesmo quando todos os DCs pertencem à mesma nuvem, cada DC pode operar em um segmento de rede diferente. Neste caso, deve-se considerar que existam diferentes políticas de segurança e gerenciamento, além de configurações técnicas distintas.

Para descrever o modelo de gerência da migração de VMs, são descritos: o contexto em que migrações são executadas; as ações de gerenciamento necessárias em cada etapa do ciclo de vida de uma migração e também na execução do plano de migração e as políticas adotadas em cada etapa. Além disso, o modelo detalha as opções de condução dos diferentes fluxos durante o planejamento e a execução de uma migração.

As políticas e algoritmos usados no planejamento de migrações apresentadas no Capítulo 4 podem ser reduzidas a duas categorias: políticas para a seleção de VMs migrantes e para a escolha do *host* de destino. Ambas são essenciais para a construção de mapas de migração que compõem o planejamento da migração. Contudo, alguns conceitos são essenciais para que sejam descritas as políticas utilizadas nesta tese. Eles são descritos a seguir.

A *capacidade* do recurso  $r$  no *host*  $h$ , ou seja, a quantidade de  $r$  presente em  $h$ , é dada por  $C(h, r)$ . Sendo  $U(m, r)$  a quantidade do recurso  $r$  usado pela VM  $m$  e  $M(h)$  o conjunto de VMs no *host*  $h$ , o uso atual de um recurso  $r$  no *host*  $h$  é definido na Equação 5.1.

$$U(h, r) = \sum U(m, r) \forall m \in M(h) \quad (5.1)$$

Como um *host* não pode prover mais recursos do que possui,  $0 \leq U(h, r) \leq C(h, r)$ , a *capacidade ociosa* de um recurso  $r$  em  $h$ , ou seja, a quantidade do recurso  $r$  livre para novas VMs em  $h$ , é dada por  $A(h, r) = C(h, r) - U(h, r)$ . Por fim, a *demand* do recurso  $r$  pela VM  $m$  ou a quantidade de  $r$  que foi contratado para a VM  $m$  é dada por  $D(m, r)$ . Em situações normais, o uso do recurso  $r$  é idêntica à demanda contratada para a VM, ou seja,  $D(m, r) = U(m, r)$ , no entanto, nem sempre uma VM usará o que foi contratado ou até mesmo, em situações de instabilidade, a VM pode tentar usar mais do que foi contratado.

Migrações são analisadas na ocorrência de um dos seguintes eventos: a sobrecarga de um *host* ou sua subutilização. O primeiro caso é designado como *hot spot* e o outro como *cold spot*. Em ambos os casos, a capacidade de processamento utilizada pelas VMs do *host* determinam a necessidade de migrar. Por esse motivo, nesta tese, tanto a capacidade ociosa como a demanda estão baseadas no uso do processador. Como o consumo do processador é o recurso mais relevante nas decisões de alocação e migração, pode-se simplificar a notação e afirmar que  $C(h) \triangleq C(h, cpu)$ ,  $U(h) \triangleq U(h, cpu)$ , e assim por diante.

## 5.2 IDENTIFICANDO A NECESSIDADE DE MIGRAÇÕES

Comprovar uma necessidade de migração envolve a detecção de um gatilho de migração. Neste trabalho, o gatilho considerado é a sobrecarga de *host* (*hot spot*). Tal tipo de gatilho pode ser indício de diferentes condições que exijam atenção, tais como: aumento do consumo de energia, incremento da dissipação de calor, ocorrência de violações de SLAs e necessidade de manutenção (usualmente, em decorrência de alguma falha). Ou seja, localizar gatilhos de migração permite tomar decisões para solucionar diferentes situações anômalas.

A fim de evitar que a detecção dos gatilhos de migração seja baseada em uma situação momentânea, por exemplo, um pico de uso de um ou mais recursos, é usada a técnica conhecida como janela de tempo deslizante. Ou seja, averiguações de *thresholds* são feitas em todo intervalo de tempo, mas, a decisão por planejar ou não uma migração ocorre no término de cada janela de tempo deslizante.

As janelas de tempo deslizantes são utilizadas para permitir que uma necessidade de migração seja confirmada. Dois aspectos envolvem a decisão: o tamanho da janela de tempo deslizante ( $W$ ) e a quantidade de eventos que confirmariam a necessidade de migrar ( $wThr$ ), ou seja, os gatilhos de migração. O tamanho de ambas influencia na identificação de uma necessidade de migração e, conseqüentemente, no seu sucesso. Se  $W$  é um intervalo pequeno, tem-se detecções mais rápidas, tendendo a ignorar eventos transientes, tais como picos momentâneos de uso dos recursos. De outra forma, se  $W$  é um intervalo maior, tem-se detecções mais lentas, contudo com uma melhor avaliação das potenciais transiências (Wood et al., 2007; Luiz et al., 2010; Xu e Fortes, 2011). Este dilema também existe para a definição de  $wThr$ . *Thresholds* menores conduzem a uma abordagem de execução agressiva de migrações, ou seja, há maior constância da tarefa de planejar e executar migrações. Por outro lado, o valor para  $wThr$  maior indica uma relativa permissividade, com o risco de que haja mais potenciais violações de SLA.

Há abordagens para a definição dinâmica dos valores de  $W$  e  $wThr$ . Neste caso, dados históricos de uso dos recursos pelos *workloads* permitem traçar o potencial de transiência. Para

Beloglazov e Buyya (2010), *thresholds* com valores fixos não refletem o comportamento dinâmico dos *workloads* e, por consequência, tendem a gerar decisões de migração não satisfatórias. Chaabouni e Khemakhem (2018) ressaltam a relevância dos *thresholds* dinâmicos, destacando que eles têm o potencial adequado para atender aos requisitos dos consumidores dos CSs.

Neste trabalho, o dinamismo está no intervalo de tempo adotado para avaliar a necessidade de migrar. Ou seja, o valor  $wThr$  é fixo, mas, o valor de  $W$  varia conforme o comportamento do ambiente. De fato, trabalhos anteriores consideraram *thresholds* fixos para o consumo seguro de recursos variando de 70% a 90% (Wood et al., 2007; Xu e Fortes, 2011; Kollberg et al., 2020). A escolha dos valores, nos trabalhos anteriores, foi empírica, considerando que o valor produziria resultados satisfatórios aos propósitos de cada trabalho, o que pode ser confirmado nas respectivas avaliações. Considerando esse panorama, nesta tese o valor de *threshold* escolhido foi um valor menos propenso à permissividade de violações de SLAs do que pode ser observado nos trabalhos relacionados: 75%. Ou seja, se uma quantidade de violações ocorrer em mais de 75% dos intervalos, considera-se que a instabilidade não foi momentânea e que, de fato, é necessária uma migração.

Considerando que *workloads* têm comportamento dinâmico, variando a quantidade de recursos em uso, decidiu-se que as janelas deslizantes não sejam fixas. Para isso, conforme ilustrado na Figura 5.1, duas medidas são avaliadas em intervalos de janelas deslizantes:  $W_{ws}$  e  $W_{ev}$ . A primeira controla o instante em que a quantidade de diferentes coletas de dados são avaliadas e a segunda determina o instante em que os tamanhos de janela devem ser revistos.

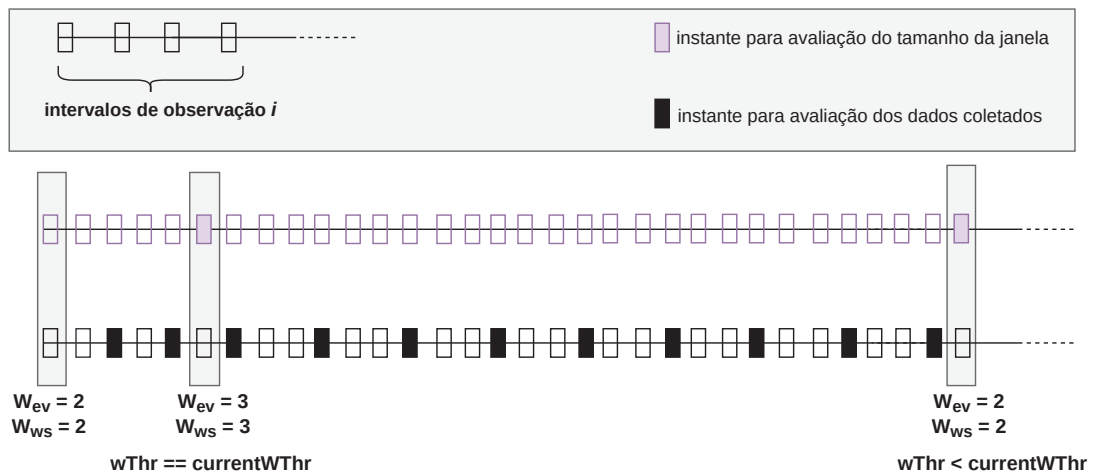


Figura 5.1: Esquema adaptativo de definição da duração das janelas deslizantes.

A situação de exemplo, apresentada na Figura 5.1, demonstra que no início das avaliações, ambas as janelas têm tamanho 2. Ou seja, após duas avaliações das coletas de dados é que o tamanho de  $W_{ev}$  é novamente reavaliado. Essa reavaliação é descrita no Algoritmo 1 que considera a existência de uma variável  $wThr$  que armazena a quantidade de violações de *thresholds* durante todas as  $W_{ev}$  iterações. Considera-se que um comportamento estacionário dos dados, ou seja, mantendo a mesma utilização dos recursos, não incorra na necessidade de alterar o tamanho das janelas. Contudo, caso ocorra alguma alteração deve-se ponderar entre adotar janelas mais restritivas (menores) ou maiores.

---

**Algoritmo 1** Algoritmo para o cálculo das janelas deslizantes.

---

```

1: if  $wThr == -1$  then
2:    $W_{ev} = 2$ 
3:    $W_{ws} = 2$ 
4: else
5:   if  $wThr > 0$  then
6:     if  $wThr > currentWThr$  then
7:       incrementa o valor de  $wThr$ 
8:     else
9:       if  $wThr < currentWThr$  then
10:        decrementa o valor de  $wThr$ 
11:      end if
12:    end if
13:  end if
14:   $W_{ev} = wThr$ 
15:   $W_{ws} = wThr$ 
16: end if

```

---

A cada encerramento de uma janela  $W_{ws}$ , deve-se coletar a quantidade de violações de *threshold* ocorrida em cada *host*. *Hosts* com violações em mais de 75% dos intervalos de uma janela são incluídos em uma lista de *hosts* instáveis (*wInstList*).

Para cada *host* na lista *wInstList* são disparados pedidos de planejamento de migrações. Em caso de serem identificados *hosts* na lista *wSubList*, deve-se prover a migração das VMs de cada *host* para outros *hosts* disponíveis (balanceamento de carga).

Recursos requeridos por migrações em curso devem ser considerados nas coletas de dados. Esse cômputo na distribuição dos recursos pode advir da reserva de recursos. Contudo, embora a reserva de recursos seja uma etapa natural de uma migração a quente, ela é destinada a cenários de uma rede local e restritos à migração de uma única VM (Clark et al., 2005). Ou seja, mesmo em um cenário intra-DC, é possível que uma VM migrante ainda não tenha sido definitivamente alocada no novo *host*, gerando discrepâncias na coleta de dados.

Estratégias para a reserva de recursos em ambientes de múltiplas nuvens já foram propostas por (Biran et al., 2018; K.Ye et al., 2011). Em decorrência, diferentes modelos para o planejamento da capacidade de nuvens e também para a predição de carga têm sido propostos. No entanto, esse detalhamento desvia do propósito desse trabalho, pois requer uma discussão que envolve mais do que o trabalho de gerenciamento dos recursos de uma nuvem, mas, também da infraestrutura e das tecnologias utilizadas. Por esse motivo, no tocante à coleta de dados, é considerado que reservas de recursos para quaisquer VMs migrantes já tenham sido providenciadas, estando portanto, à disposição no momento em que cada coleta é feita.

### 5.3 SELEÇÃO DE VMS A MIGRAR

Dois propósitos podem ser identificados em políticas para a escolha de VMs migrantes. Em um grupo estão as políticas que têm como principal interesse o menor custo da migração. Nesse grupo estão estratégias tais como a de Wood et al. (2007), que usa a métrica VSR. Esta métrica captura a relação entre o volume de uma VM (V) e a sua quantidade de memória em uso. Essa relação permite escolher VMs que imporão um tráfego mais leve e, portanto, gerando migrações mais rápidas. Pode-se citar as políticas sugeridas por Beloglazov e Buyya (2012), cujos propósitos também são diminuir o tempo para a execução de uma migração. No outro grupo estão as iniciativas que focam na eficiência da manutenção dos padrões de desempenho da VM



que migrará. Esse é o caso dos trabalhos de (Mashtizadeh et al., 2014; Basu et al., 2019), cuja escolha de VMs migrantes tem como princípio a diminuição do *downtime* dos CSs associados às VMs, ou seja, o interesse principal é garantir que o consumidor dos CSs mantenham a mesma percepção do desempenho, apesar da migração.

Aliar ambos os propósitos de seleção de VMs migrantes mencionados é uma estratégia que, até onde se pode averiguar, ainda não foi considerada. Consideramos que aliar ambos tem o potencial de gerar um conjunto reduzido de opções de migração que, no entanto, sejam efetivas em ambos os âmbitos: na minimização do custo da migração e na manutenção dos níveis de desempenho da VM. Um universo reduzido de opções diminui a complexidade para se localizar boas soluções além de diminuir o tempo usualmente gasto para o planejamento de uma migração.

Em caso de correção da instabilidade por conta de *hot spots*, a lista de *hosts* instáveis (*wInstList*) é ordenada em ordem decrescente conforme o uso da capacidade de processamento, gerando uma nova lista (*cpuHList*). Para cada *host* ( $h_{ov}$ ) em *cpuHList*, planos de migração precisam ser construídos. O primeiro passo é a escolha de quais VMs podem, de fato, participar de uma migração. As VMs de  $h_{ov}$  que já estão participando de uma migração são desconsideradas, gerando, portanto uma lista de VMs que são candidatas a migrar, ou de VMs migrantes (*migratingVMList*).

Uma nova métrica foi idealizada, visando diminuir o impacto da migração e ao mesmo tempo prever garantias de que a VM manterá o seu nível desempenho. A VM que possibilita o transporte da maior quantidade de dados em menos tempo ( $V_{lowT}$ ) permite classificar o conjunto de VMs migrantes (*migratingVMList*) conforme a razão entre o volume  $V$  e a expectativa do tempo gasto para a migração da VM ( $t_m(m)$ ).

O tempo  $t_m(m)$  necessário para migrar uma VM  $m$  pode ser estimado como a relação entre o uso atual de memória  $U(m, ram)$  e a largura de banda da rede  $A(h, bw)$  disponíveis no *host*  $h$  que contém  $m$ :  $t_m(m) = U(m, ram) / A(h, bw)$ . Ou seja, o  $V_{lowT}$  de uma VM  $m$  é dado por:

$$V_{lowT}(m) = \frac{V(m)}{t_m(m)} \quad (5.2)$$

As VMs da lista *migratingVMList* são organizadas em ordem decrescente conforme a métrica descrita na Equação 5.2. Consideramos que uma VM que possui largura de banda disponível suficiente para transportar uma quantidade maior de dados é a que causará menor impacto no custo e também que tem maiores chances de manter a estabilidade da oferta dos serviços associados à VM. De fato, se o tempo médio para transportar os dados da VM é menor, então a possibilidade de ocorrência de *downtime* também é menor, conforme descrito por (Voorsluys et al., 2009; Strunk, 2012).

#### 5.4 SELEÇÃO DE HOSTS DE DESTINO DE MIGRAÇÕES

Diferentes opções de migração podem ser geradas, resultando em tuplas do tipo  $\langle m, h_{dest} \rangle$ , onde  $m$  é uma VM candidata a migrar e  $h_{dest}$  um *host* de destino que seja capaz de atender às necessidades de recursos da VM  $m$ . Na lista de candidatos a destino para a VM migrante devem constar apenas os que possuem capacidade ociosa para receber  $m$ , ou seja se  $A(h) > D(m)$ .

Além da verificação da disponibilidade de recursos, outra métrica contribui para tornar os mapas de migração efetivos: a previsão de demanda da VM  $m$ . Conhecer o quanto de recursos a VM candidata a migrar, de fato, exigirá após a migração contribui em dois aspectos. Primeiro, para confirmar que o *host*  $h_{dest}$  poderá atender às necessidades da VM e também contribui porque antecipa-se à possibilidade de um novo episódio de instabilidade.



Embora seja possível identificar a demanda corrente da VM no instante em que a necessidade de migração tenha sido verificada, é necessário prever  $D(m, r)$  no instante em que um  $h_{\text{dest}}$  a recebe. Essa ação contribui para garantir que a migração será efetiva. Ou seja, o valor previsto de demanda da VM  $m$  ( $D(m, r)_{t+1}$ ) permite confirmar que a escolha de  $h_{\text{dest}}$  contribui para que a instabilidade de  $h_{\text{ov}}$  seja solucionada.

Um modelo de auto-regressão linear similar ao proposto por Wood et al. (2007) é usado, considerando dados sobre o uso dos recursos coletados, conforme descrito na Figura 5.1. Para que se considere uma efetiva tendência do comportamento dos dados coletados, duas suposições são feitas: o comportamento dos dados de uso dos recursos apresenta um crescimento positivo linear e a previsão de demanda só ocorre a partir do décimo intervalo. Pode-se observar, dos experimentos iniciais, que a partir da décima coleta de dados, o comportamento das VMs de um *host* apresentavam menor variação de uso dos recursos (quando ocorriam), por isso a escolha de se iniciar a previsão de demanda apenas a partir da décima coleta de dados. Antes disso, a demanda prevista é considerada pelo cálculo da média dos dados previstos. Para identificar a demanda  $D(m, r)_{t+1}$ , considera-se que os processos auto-regressivos de ordem  $n$  ( $AR(n)$ ) usem as  $n - 1$  observações anteriores. Dessa forma, a previsão de demanda da VM  $m$  é dada por:

$$D_{t+1} = \mu + \phi(D_t - \mu) \quad (5.3)$$

onde  $\mu$  representa a média de  $U(m, r)$  já coletada e  $\phi$  é uma função que permite ajustar a série de valores coletados em relação a possíveis variações no comportamento da série (*lags*).

Para que planos de migração sejam gerados, é necessária a definição de uma estratégia para a escolha de potenciais *hosts* de destino. De maneira simplificada, uma estratégia imediata para a escolha de um destino é considerar que haja disponibilidade de recursos para suportar a VM migrante, ou seja:  $A(h_{\text{dest}}, r) > D(m, r)_{t+1}$ . No entanto, essa condição não considera o cenário real de  $h_{\text{dest}}$ , estando propenso a gerar uma nova instabilidade. Por esta razão, mapas de migração precisam atender ao requisito descrito como Efetividade ( $Ef(< m, h >)$ ). Considera-se que um mapa de migração  $< m, h >$  propõe uma migração satisfatória quando o potencial de eventos indesejados seja pequeno. Situações como a indisponibilidade de recursos no *host*, sobrecarga súbita do *host* ou até mesmo da VM tornam a migração sem efeito, não solucionando a instabilidade de oferta de recursos observada. A Equação 5.4 descreve que um destino só seja incluído em um mapa de migração se não houver o potencial de sobrecarga do *host* de destino. Dessa forma, tem-se que um mapa de migração ( $< m, h >$ ) somente pode ser inserido em um plano de migração se sua efetividade  $Ef(< m, h >)$  for verdadeira.

$$Ef(< m, h >) = D(h_{\text{dest}}, r) + D(m, r)_{t+1} < 0,75 * (A(h_{\text{dest}}, r)) \quad (5.4)$$

O Algoritmo 2 descreve as ações para a localização de um *host* que atenda ao requisito mencionado. É importante salientar que recursos podem ser procurados em outros DCs, caso não possa ser atendido intra-DC. Contudo, para que outros DCs sejam procurados em outras nuvens, considera-se que existam acordos prévios prevendo essa possibilidade.

---

**Algoritmo 2** Algoritmo para a identificação de potenciais  $h_{\text{dest}}$ .

---

```

1: function findDest(m)
2: faça hDCList a lista de todos os hosts do mesmo DC de  $h_{\text{ov}}$ 
3: faça currentTargetList a lista de todos os hosts disponíveis para a migração
4: currentTargetList = null
5: for all  $dc \in DCList$  do
6:   for all  $h_{\text{dest}} \in hDCList$  do
7:     if  $AU(h_{\text{dest}}, r) > D(m, r)_{t+1}$  then
8:       adicionar  $h_{\text{dest}}$  a currentTargetList
9:     end if
10:   end for
11: end for
12: if currentTargetList == null then
13:   buscar opções de destino em outras nuvens
14:   for all  $cm \in CMPartnerList$  do
15:     for all  $dc \in DCList$  do
16:       for all  $h_{\text{dest}} \in hDCList$  do
17:         if  $A(h_{\text{dest}}, r) > D(m, r)_{t+1}$  then
18:           adicionar  $h_{\text{dest}}$  a currentTargetList
19:         end if
20:       end for
21:     end for
22:   end for
23: end if
24: return currentTargetList

```

---

O Algoritmo 3 descreve as ações necessárias para criar todos os planos de migração do ambiente. Para sua execução, considera-se que *findDest(m)* retorne uma lista de destinos que estejam preparados para receber. Essa lista, no entanto, pode conter *hosts* cuja participação em um migração pode não gerar resultados satisfatórios, tais como a sobrecarga eminente. Ou seja, a inclusão do *host* em um mapa de migração ocorre apenas após esta averiguação.

---

**Algoritmo 3** Algoritmo para a criação de planos de migração.

---

```

1: defina planMigList o plano de todas as migrações
2: faça planOrder o indexador de cada host
3: planMigList = null
4: for all  $h_{ov} \in wInstList$  do
5:   planOrder ++
6:   for all  $minh_{ov}$  do
7:     if  $m$  não é migrante then
8:       adicionar  $m$  em migratingVmList
9:     end if
10:  end for
11:  ordenar migratingVmList conforme  $V_{lowT}(m)$ 
12:  for all  $m \in migratingVmList$  do
13:     $hDestList = findDest(m)$ 
14:    for all  $h_{dest} \in hDestList$  do
15:      if  $Ef(< m, h_{dest} >)$  then
16:        adicionar  $< m, h_{dest} >$  a planMigList[planOrder]
17:      end if
18:    end for
19:  end for
20: end for
21: return planMigList

```

---

## 5.5 CONSIDERAÇÕES FINAIS

Migrações quando executadas isoladamente por uma Plataforma de Nuvem ou por um Sistema Operacional de Nuvem tendem a ser prejudicadas pela quantidade de iterações necessárias para lidar com o planejamento das migrações. Também são prejudicadas por terem pouca capacidade de acessar informações sobre o andamento das migrações. Por tais razões, uma solução de gerenciamento de migrações demonstra o potencial de diminuir os impactos negativos de se lidar com migrações de forma isolada e dependente de operadores humanos.

Estratégias concebidas no modelo permitem que os resultados de uma migração sejam ainda mais positivos. Por exemplo, prever a efetividade de uma migração diminui o potencial de se gerar migrações mal sucedidas que desperdiçam esforço para seu planejamento e também o tempo para sua execução. É importante ressaltar que a análise da transiência de um evento de migração também contribui para diminuir o potencial de ocorrência de migrações mal sucedidas.

Um ambiente de nuvem é uma entidade dinâmica, dependendo das parcerias previamente estabelecidas pelo CSP. É importante ressaltar que as parcerias estabelecidas ao nível do CP não são consideradas porque entre CSP e CP é que estão estabelecidos os acordos, e consequentemente, objetivos e restrições, associados ao CS e, portanto, às VMs que suportam a provisão do serviço.

O gerenciamento de migrações inter-nuvem esbarra em obstáculos que dependem de tecnologias específicas, tais como a criação de um canal privado ou de adaptações na estruturação de VMs. Esses aspectos dificultam a apresentação de uma solução de gerenciamento inter-nuvem abrangente. No entanto, a busca por recursos pode ser estendida dinamicamente pela procura de recursos nos parceiros já estabelecidos pelo CSP.

## 6 UMA ARQUITETURA PARA A GERÊNCIA DE MIGRAÇÕES

O modelo apresentado no Capítulo 5 pode ser empregado em distintos cenários, tais como: na integração de um CSP a uma nuvem híbrida (virtualização horizontal); no controle dos recursos disponíveis em um DC e ainda para que um CP estenda sua capacidade de recursos, gerando o potencial de CSPs idealizarem novos serviços (virtualização vertical). Embora sejam cenários distintos, é possível supor uma arquitetura que implemente o modelo apresentado e que seja adequada a todos esses cenários.

Neste capítulo a arquitetura para a gerência de migrações é apresentada, sendo inicialmente discutida na Seção 6.1. Os componentes da arquitetura, suas interfaces e a integração do modelo é apresentada na Seção 6.2 e, ao final, são apresentadas, na Seção 6.3, as considerações finais.

### 6.1 VISÃO GERAL DA ARQUITETURA

CSPs e CPs usam a migração de VMs para prover a alocação dinâmica de recursos. Em geral, ambas as entidades dependem da estrutura de planejamento e execução de migrações oferecida pelo Provedor de IaaS e que, em muitos casos, não têm acesso a informações dos CSs atualizadas sobre sua provisão. É natural que a concentração de uma diversidade de planos de migração sobrecarreguem os serviços de automação de migração integrados a CMPs ou a Sistemas Operacionais de Nuvem, requerendo uma solução para a gerência de migrações. A solução para o gerenciamento da migração de VMs apresentada nesta tese, oferece a ambos (CSPs e CPs) uma estratégia para a garantia do dinamismo na oferta de recursos, aliviando o esforço que as atuais abordagens de automação de migração requerem.

A gerência da migração de VMs deve permitir que instabilidades na oferta de recursos sejam solucionadas. Para que isso ocorra, três níveis de informação precisam ser coordenados: (a) infraestrutura; (b) virtualização e (c) controle. A infraestrutura é designada às operações executadas por sistemas operacionais e sistemas de gerenciamento de redes e serviços. No nível de virtualização estão as ações executadas pelos hipervisores, tais como criação, destruição e migração de VMs. Entretanto, é no nível de controle que o mapeamento dos recursos virtuais das VMs é gerenciado. Entidades como CSPs, CPs e Provedores de IaaS executam as ações neste nível, demandando estratégias para gerenciar migrações sob diferentes perspectivas.

Neste capítulo é apresentada uma arquitetura de software distribuída que implementa o modelo proposto no capítulo anterior. Essa arquitetura é constituída dos seguintes componentes: Gerenciador da Nuvem (*Cloud Manager* - CM), componente responsável por coordenar as tarefas inter-DC, especialmente as migrações que não podem ocorrer no domínio de um único DC, sendo executado no domínio de uma nuvem. Gerenciador do *Datacenter* (*Datacenter Manager* - DCM), responsável pelo gerenciamento dos recursos de um DC, podendo conduzir migrações intra-DC e sendo executado no âmbito de um DC. Controlador do *Host* (*Host Controller*), componente que interage com o hipervisor do *host* e que coleta, periodicamente, dados das VMs alocadas em seu domínio e reportando ao DCM gatilhos que podem tornar-se migrações efetivas. Broker é o componente que permite aos consumidores requisitarem a criação de novas VMs assim como submeterem *workloads* para serem executados.

## 6.2 DETALHAMENTO DOS COMPONENTES E SUAS INTERAÇÕES

Na Figura 6.1 é apresentada a visão funcional da arquitetura de gerenciamento de migrações. O HC é um componente presente em todo *host* provendo interfaces para comunicação com o hipervisor e também para comunicação com o DCM que também provê interfaces para comunicação com cada HC. Cada CM tem interfaces para cada DCM, permitindo, portanto, o planejamento de migrações inter-DC dentro de uma mesma nuvem.

No tocante às interfaces, o Gerenciador do DC (*DC Manager* - DCM). O DCM pode ser integrado a diferentes ferramentas de gerenciamento da nuvem, tais como CMPs, o Sistema Operacional da Nuvem e serviços executados ao nível do DC, tais como: orquestradores de CSs e ferramentas de automação da migração.

Os recursos físicos que provêm a infraestrutura dos *hosts* são coordenados pelas ferramentas de gerenciamento dos respectivos ambientes, provendo ao Controlador de *Host* (*Host Controller* - HC), descrito na Figura 6.1, a capacidade de virtualizar tais recursos. As VMs, gerenciadas pelo hipervisor associado ao HC são usadas na provisão de CSs e são a fonte de informação sobre o uso, a demanda e a disponibilidade dos recursos virtuais.

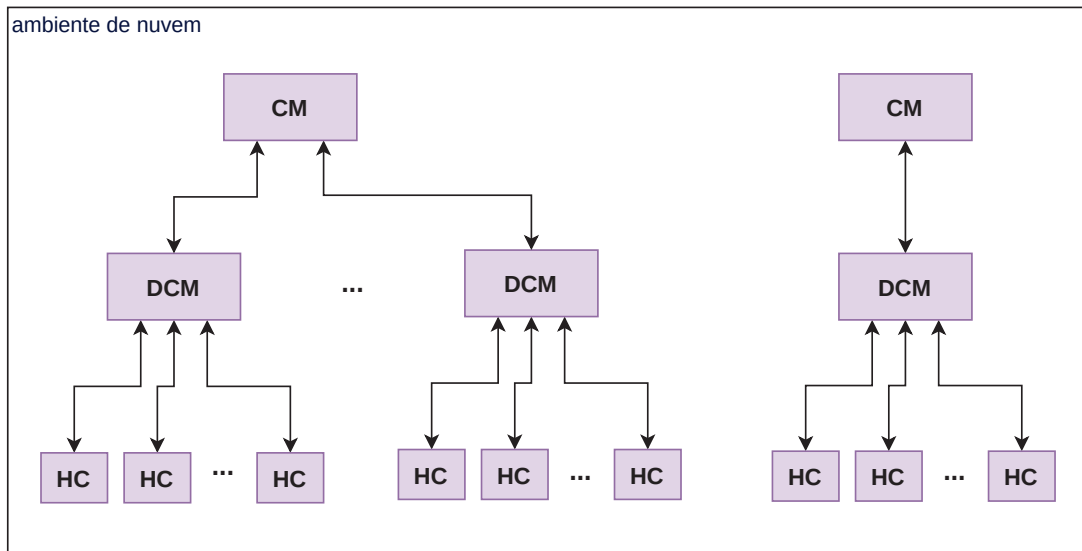


Figura 6.1: Arquitetura para o gerenciamento da migração de VMs.

Migrações são, primariamente, identificadas pelos próprios componentes do ambiente da nuvem. Todo HC inspeciona periodicamente o uso dos recursos de cada VM gerenciada pelo seu hipervisor local, registrando em um repositório do DC, o resultado da coleta de  $U(m, r) \forall m \in M(h)$ . Ao término de uma janela  $W_{ws}$ , o DCM tem a capacidade de identificar quais *hosts* atingiram ou ultrapassaram um *threshold* definido, ou seja em quais *hosts*  $U(h, r) \geq T(h, r)$ . Tais *hosts* passam a ser considerados instáveis, requerendo o disparo do planejamento de migrações em cada um. Conforme descrito no Capítulo 5, considera-se nesta tese que *thresholds* para a detecção de instabilidade de *hosts* são de 75%.

Na Figura 6.2 são destacadas três interfaces: uma que permite interação com o DCM, outra com a infraestrutura do *host* associado e também uma interface com o repositório que armazena as coletas periódicas dos dados das VMs. A primeira interface lida com atividades associadas aos níveis de controle e virtualização, a segunda é designada ao gerenciamento da infraestrutura e dos recursos (físicos e virtuais) e a última atua como uma forma de integrar todos HCs e o DCM.

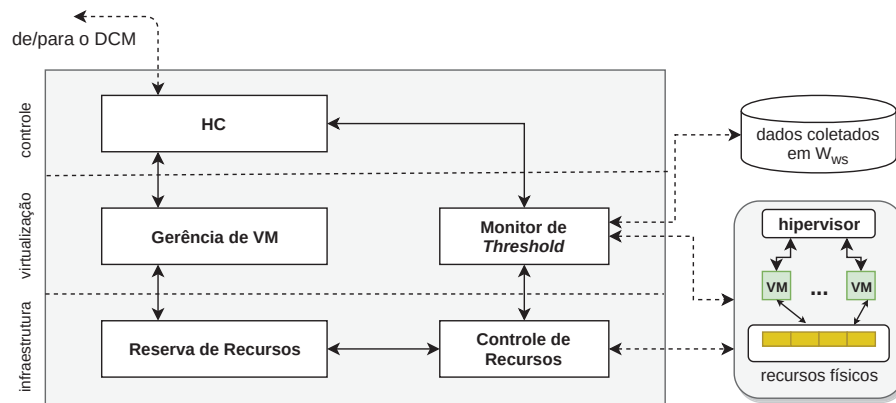


Figura 6.2: Operações e interfaces do Controlador de *Hosts* (HC).

Ao HC é destinada a responsabilidade da coleta periódica de dados que reflitam o uso dos recursos. O DCM define os intervalos em que tais coletas devem ocorrer, indicando ao HC que a operação deve ser feita. Esse procedimento é descrito na Figura 6.3, ressaltando que a cada coleta executada, o resultado deve ser registrado em um repositório de dados compartilhado com o DCM.

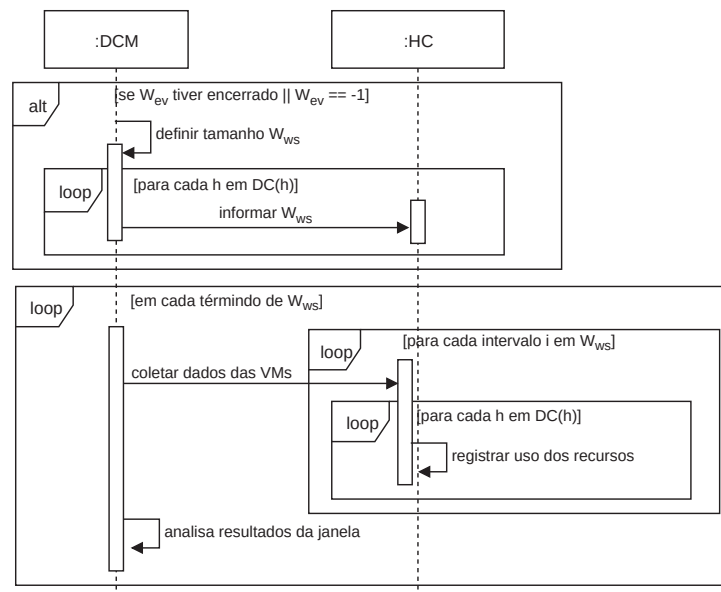


Figura 6.3: Interações entre DCM e HC para a coleta de dados de VMs.

O DCM concentra as ações para a elaboração de planos de migração, definindo, a cada ciclo de  $W_{ev}$  iterações, um novo valor de  $W_{ws}$  para os respectivos *hosts*. Além de atender a diferentes interfaces para outros componentes da arquitetura (HC e CM), também provê interações com usuários que podem ser CSPs, CPs e também componentes de software dedicados ao gerenciamento de nuvens, tais como: os Sistemas Operacionais de Nuvens e CMPs.

Conforme descrito na Figura 6.4, migrações são planejadas para cada DC. É importante ressaltar que as políticas definidas no Capítulo 6 determinam a ordem em que devem ser tentadas as migrações. Além dessa decisão, também deve-se garantir se a migração possibilita, de fato, resultados satisfatórios. Para esse propósito, é necessário prever a demanda da VM. Para essa operação, o DCM precisa identificar que *hosts* são potencialmente adequados para hospedar a VM migrante. Via de regra, esses parceiros são buscados no âmbito do próprio DC. No entanto,



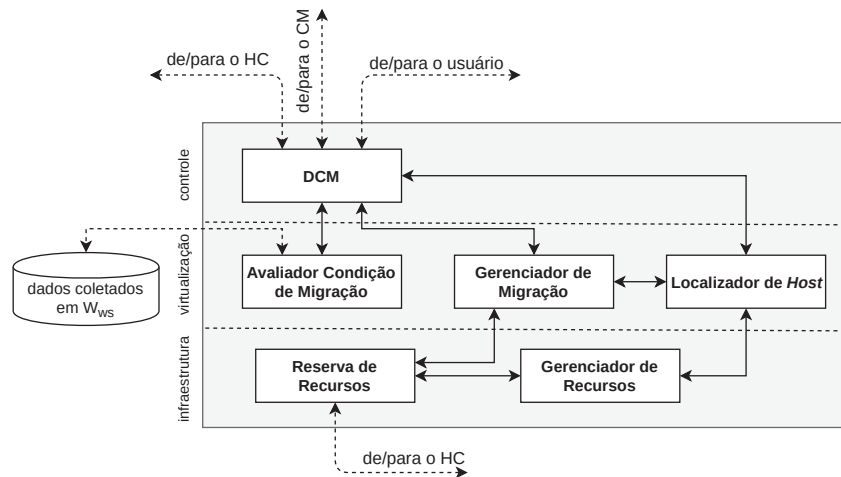


Figura 6.4: Operações e interfaces do Gerenciador de DC (DCM).

caso não haja disponibilidade efetiva em nenhum parceiro local, a busca é direcionada para o componente CM.

Como planos de migração podem ser compostos de múltiplas alternativas de migração para uma VM, tão logo se confirme a efetividade da migração, o mapa  $\langle m, h_{dest} \rangle$  só é incluído no plano após os recursos serem reservados no *host* de destino. Portanto, como descrito na Figura 6.5, na execução de um plano de migração, é necessário considerar a necessidade de liberar recursos que não foram utilizados, já que algum mapa de migração já atendeu à necessidade.

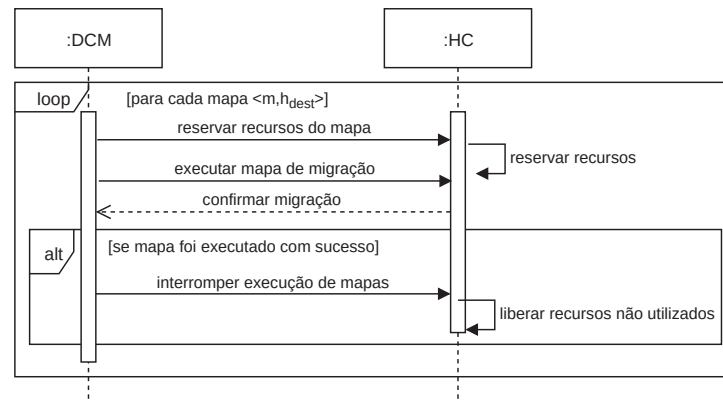


Figura 6.5: Execução de um plano de migração com recursos já reservados.

O Gerenciador da Nuvem (CM) atua principalmente na descoberta de recursos intra-nuvem. Outros DCs da mesma nuvem podem ser contactados, via seus respectivos DCMs, conforme descrito na Figura 6.6. Necessidades de recursos que não podem ser resolvidas intra-nuvem somente podem ser solucionadas inter-nuvem para ambientes com quem o CSP mantém parcerias. Em geral, isso pode significar a "explosão" do ambiente para uma nuvem híbrida. Detalhes de como a estruturação desse novo ambiente é executada fogem ao escopo deste trabalho.

Como apresentado na Figura 6.7, migrações inter-DC somente são planejadas apenas se a necessidade de migração não é solucionada no âmbito intra-DC. Considerando que os mapas que compõem o plano estão organizados conforme as melhores opções para a solução da oferta de recursos, tão logo o primeiro mapa for executado com sucesso, deve-se cancelar a execução das outras opções, liberando os recursos que foram previamente reservados.

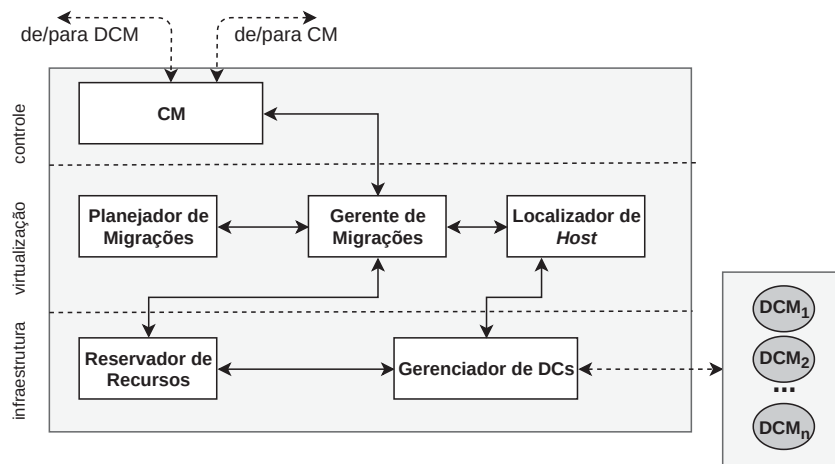


Figura 6.6: Operações e interfaces do Gerenciador de Nuvem (CM).

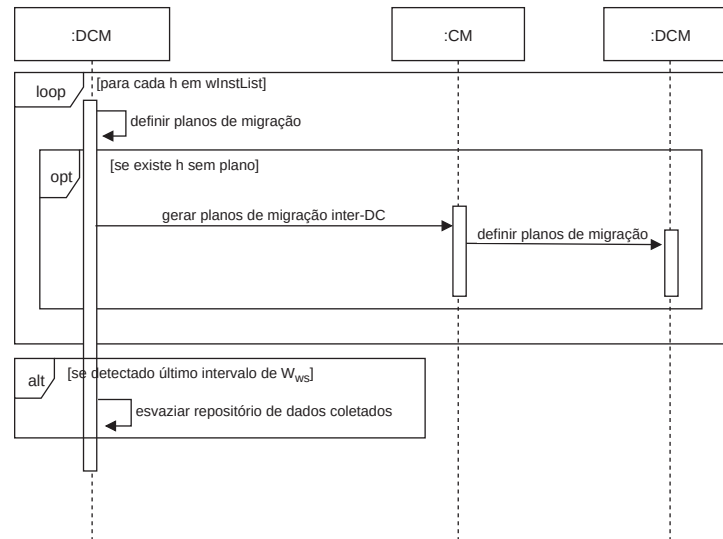


Figura 6.7: Operações para a geração de planos de migração.

### 6.3 CONSIDERAÇÕES FINAIS

A arquitetura apresentada promove soluções para a oferta dinâmica de recursos em diferentes cenários: intra-DC, inter-DC (intra-nuvem) e inter-DC (inter-nuvem). No entanto, a arquitetura não lida com cenários dinâmicos, considerando que um ambiente de nuvem com acordos já pré-estabelecidos seja considerado.

O componentes da arquitetura possibilitam que, através da migração, um CSP possa, a partir dos acordos previamente firmados, alargar o domínio de seus recursos que, prioritariamente, estavam reduzidos ao DC em que a VM associada ao seu CS estava hospedada. Para que isso ocorra, diferentes CMs devem ser implementados em nuvens distintas, permitindo a comunicação da necessidade de migração.

Para que a adaptabilidade dos intervalos de avaliação de recursos possa ser considerada uma contribuição no planejamento de migrações efetivas, ações entre DCM e HC são imprescindíveis, permitindo que novos intervalos de avaliação sejam concebidos e divulgados. É importante salientar que ao fim de cada intervalo de avaliação de *thresholds* existe um intervalo  $i$  para que o DCM possa acessar dados do repositório de coletas atualizados.

Coletas de dados após terem sido processados pelo DCM são excluídas do repositório. Essa decisão permite que o DCM lide sempre com dados da janela em curso. Contudo, cada DCM mantém um histórico das coletas de cada HC de seu domínio, podendo ser usado para decisões como a demanda futura de recursos de uma VM.

A reserva de recursos é tarefa essencial para garantia da estabilidade das decisões de migração, no entanto, ela estimula um dinamismo no mapeamento de recursos virtuais aos físicos. Recursos que foram reservados precisam ser liberados, caso contrário, alocações podem ser comprometidos. Por esse motivo, na execução de planos, tão logo um mapa de migração tenha sucesso, deve-se liberar recursos associados aos mapas que não precisaram ser executados.

## 7 AVALIAÇÃO

Este capítulo apresenta a avaliação do modelo para a gerência da migração de VMs e sua respectiva proposta de implementação, apresentada no Capítulo 6. Na Seção 7.1 são descritas características da avaliação, baseada em simulações e do ambiente em que estas foram executadas. Na Seção 7.2 é apresentada uma descrição de como a arquitetura é simulada no ambiente descrito na seção anterior. Posteriormente, na Seção 7.3 é descrito como as simulações são avaliadas, permitindo descrever como os resultados dos experimentos foram alcançados. Na Seção 7.4 são apresentados os resultados do primeiro experimento que avalia o sucesso das migrações planejadas em dois cenários distintos: baseado no uso de ações de gerenciamento baseados na arquitetura apresentada e não baseadas em tais ações. A Seção 7.5 apresenta uma avaliação da sobrecarga dos *hosts*, considerando que a continuidade de sobrecargas indica insucesso na execução de planos de migração.

### 7.1 INTRODUÇÃO

Espera-se que as decisões norteadas pela arquitetura de migrações de VMs conduzam a planos de migração mais efetivos e, conseqüentemente, execuções que não tragam prejuízos ao ambiente. Neste Capítulo, as contribuições são avaliadas, permitindo identificar se a arquitetura de gerenciamento conduz, de fato, a remapeamentos dinâmicos de recursos que sejam efetivos. É também comparado o impacto da adoção de uma modelo de gerenciamento de migrações em contraposição a uma solução não baseada em qualquer modelo.

As avaliações são conduzidas sob um modelo de simulação. Essa decisão é consequência da complexidade e da dimensão do ambiente de nuvem. Embora números precisos não sejam divulgados pelos principais atores da provisão de infraestrutura de nuvem, é possível deduzir que a extensão de ambientes de nuvens públicas e também as privadas envolvem quantidades elevadas de computadores e recursos. Outro aspecto que motiva a adoção da simulação nas avaliações é a possibilidade de abstrair aspectos específicos tais como a infraestrutura de comunicação de DCs ou a tecnologia empregada na representação das imagens das VMs. Distanciar de aspectos que não interferem na compreensão do comportamento da arquitetura ajudam na averiguação de seu desempenho, contribuindo para a avaliação das ações de gerenciamento propostas.

O simulador de ambientes de computação em nuvem *CloudSim* (Calheiros et al., 2011) foi estendido e adaptado para que a arquitetura de gerenciamento fosse avaliada. Na sua versão original, migrações não são suportadas, assim como entidades essenciais como o Gerenciador de Nuvem (CM) e o Gerenciador de DC (DCM). Também foram necessárias extensões adaptando DCs a lidarem com tarefas do gerenciamento de migrações. Para implementar a arquitetura proposta no *CloudSim*, 15 novas classes foram adicionadas ao código do simulador, 15 classes foram estendidas e 21 foram alteradas, totalizando cerca de 4.000 novas linhas de código em Java. Os modelos de teste implementados contêm cerca de 2.000 linhas de código em 10 classes distintas.

### 7.2 MODELO DE SIMULAÇÃO

O modelo apresentado nesta Seção descreve comportamentos das operações detalhadas no modelo de gerência de VMs introduzido no Capítulo 5.1. São apresentados detalhes do

comportamento dos componentes da arquitetura, das VMs, dos *workloads* e das operações de planejamento e de execução das migrações.

Para que o comportamento das ações de gerenciamento seja descrito em um modelo de simulação, é essencial que se compreenda o comportamento de VMs e *workloads* durante uma migração. Todas as decisões de migrações são baseadas em seus estados, portanto, dependendo do monitoramento de ambas.

### 7.2.1 Caracterização de VMs e Workloads

A forma como os recursos são alocados a VMs e seus *workloads* determina o comportamento das realocações. As VMs e os *workloads* podem adotar uma das seguintes políticas de alocação de recursos: *time-shared* e *space-shared*. Na primeira, cada processador pode ser usado por diferentes VMs; no entanto, cada VM pode usar o processador por um *quantum* fixo de tempo, liberando-o com a ocorrência de preempção. Na segunda política, cada VM detém um conjunto de processadores específicos, apenas liberando-os após o *workload* encerrar. Monitorar *workloads* pode se tornar uma tarefa complexa se a política de alocação *time-shared* for considerada, pois a associação entre processadores e os recursos virtuais varia frequentemente, dificultando decisões sobre violações de *thresholds*. Baseando-se nos resultados de (Tani e Amrani, 2016), neste trabalho é considerada apenas a política de alocação *space-shared*.

Os recursos de uma VM são alocados a um *workload* tão logo a simulação inicia. Recursos podem ser realocados quando da conclusão de um *workload*, mas, também podem ser alocados a um novo *workload*, criado sob demanda nas simulações. A cada instante, determinar o uso do processador por uma VM  $m$  em um dado tempo  $t$  é definido como o uso desse recurso pelo *workload*, ou seja:

$$U_t(m, cpu) = U_t(w, cpu) \quad (7.1)$$

Para avaliar a necessidade de reorganizar a oferta de recursos nos *hosts* e, consequentemente, de migrar VMs, é essencial que se conheça o consumo total de recursos do *host* em um intervalo  $i$  dentro de uma janela deslizante, ou seja  $U_i(h, cpu)$  é descrito da seguinte forma:

$$U_i(h, cpu) = \sum U_t(m, cpu) \quad \forall m \in M(h) \quad (7.2)$$

onde  $M(h)$  é o conjunto de todas as VMs de um *host*.

Em cada intervalo  $i$ , cada *host* atualiza  $U_i(h, cpu)$ . O instante em que diferentes coletas devem ser avaliadas é definido por  $W_{ws}$ , momento em que o DCM verifica, para cada *host* os dados coletados em cada intervalo  $i$ . Se a quantidade média de vezes que o *host* ultrapassou  $T(h, cpu)$  é igual ou maior ao tamanho de janela deslizante estabelecido, então, o *host* é considerado instável. Tem-se, portanto a média de violações de SLA de um *host* dentro de uma janela  $W_{ws}$  descrita como:

$$T_{Avg}(U(h)) = \frac{\sum_{n=1}^{W_{ws}} U_i(h, cpu)}{W_{ws}} \quad \forall U_i(h, cpu) \quad (7.3)$$

onde  $U(h)$  é o consumo total de recursos no *host*  $h$

Cada DCM recebe uma lista com os diferentes  $U_i(h, cpu)$  e calcula os respectivos  $T_{Avg}(U(h))$ . Cada *host*  $h$  que apresentar, ao menos, 75% de violações ou seja,  $T_{Avg}(U(h)) \leq 0.75 \times W_{ws}$ , é incluído na lista  $wInstList$ . O limite de 75% foi definido com base em outros trabalhos. Trabalhos anteriores consideraram *thresholds* para o consumo seguro de recursos variando de 70% a 90% (Wood et al., 2007; Xu e Fortes, 2011; Kollberg et al., 2020). Nesta tese

o valor de *threshold* escolhido foi um valor menos propenso à permissividade de violações de SLAs do que pode ser observado nos trabalhos relacionados: 75%. Ou seja, se uma quantidade de violações ocorrer em mais de 75% dos intervalos, considera-se que a instabilidade não foi momentânea e que, de fato, é necessária uma migração.

Após o preenchimento da lista *wInstList*, o DCM pode iniciar o planejamento das migrações. Para tanto, classifica as VMs de cada *host* presente em *wInstList*, utilizando a métrica que classifica VMs migrantes conforme sua velocidade para transferência da maior quantidade de dados, descrita na Subseção 5.3, resultando em uma lista designada *migratingVMList*. Para cada VM *m* em *migratingVMList* devem ser escolhidos *hosts* que possam atender à necessidade de oferta de recursos da VM, restabelecendo o equilíbrio do *host* de origem.

Cada *host* considerado sobrecarregado (designado como  $h_{ov}$ ) tem como resultado uma lista de vários mapas de migração, cada qual descrito como  $\langle m, h_{dest} \rangle$ . Essa lista é o plano de migrações, descrito como  $Plan(h_{ov}) = \{ \langle m, h_{dest} \rangle \}$ . Como cada *host* sobrecarregado possui seu plano de migração ( $Plan(h_{ov})$ ), por consequência, em um determinado instante *t*, o ambiente de nuvem tem um conjunto de vários planos de migração, um para cada *host* sobrecarregado, podendo ser descrito como:

$$Plan = \{ Plan(h_i) \mid 0 < i < n \wedge Plan(h_i) \neq \emptyset \} \quad (7.4)$$

Quando uma migração é efetivamente planejada, um mapa de destinos potenciais é definido. Para iniciar uma migração, o mesmo volume de recursos da VM é reservado no destino, sendo mantido estático até que a migração encerre. No *host* de origem, durante a simulação, o uso dos recursos é diminuído ao longo do tempo, em uma porcentagem baseada na largura de banda disponível para receber o tráfego no destino e também pela quantidade de dados transferida.

O domínio de recursos onde a gerência de migrações atua está restrito a uma nuvem computacional e seus respectivos *Datacenters*. O gerenciador da nuvem (CM), descrito no Capítulo 6, coordena a disponibilidade de recursos de todos os DCs.

### 7.3 MÉTODO DE AVALIAÇÃO

Trabalhos anteriores lidam com o gerenciamento de migrações sob duas perspectivas. Na primeira perspectiva o foco está em alguma ação de gerenciamento específica, como por exemplo, a automação do planejamento das migrações. A outra ocorre quando o gerenciamento depende de um ambiente ou de uma tecnologia particular. A arquitetura apresentada nesta tese resulta da integração de diferentes ações de gerenciamento, diminuindo a necessidade de intervenção humana, assim como da dependência de alguma tecnologia particular.

Por apresentar uma arquitetura baseada na integração de diferentes ações de gerenciamento, deve-se avaliar se essa integração contribui, de fato, na gerência de migrações e, especialmente, se ela propõe melhorias. Ou seja, deve-se averiguar se a escolha de determinadas técnicas e métodos em detrimento de outras escolhas apresenta melhor desempenho. Deve-se salientar também que a integração poderia não contribuir para resultados diferentes de um cenário desprovido de uma arquitetura de gerenciamento. Ou seja, deve-se avaliar se a arquitetura de gerenciamento atinge, de fato, o propósito de planejar e executar migrações, considerando que seu desempenho é melhor que um cenário não gerenciado.

Para conduzir os experimentos dois cenários são considerados: gerenciado e não-gerenciado. O cenário gerenciado é baseado em ações não previstas na solução não-gerenciada: a comprovação de violações de *thresholds*, evitando picos espúrios; a variação no tamanho de janelas deslizantes; a reserva de recursos no lado destino durante a migração; o uso de dados da execução de migrações no planejamento da migração e políticas que definem a seleção de VMs



candidatas a migrar e de *hosts* de destino (no cenário não-gerenciado é adotada uma política FCFS para tais escolhas).

Os experimentos consistem de um conjunto de simulações que usam os mesmos parâmetros, mas com valores distintos, conforme descrito na Tabela 7.1. Cada simulação é composta de rodadas. Em cada rodada são usados os mesmos valores de parâmetros tanto no esquema gerenciado como no não gerenciado. O intuito das rodadas foi obter significância estatística que permitam constatar a tendência do comportamento de cada métrica analisada. Para confirmar a consistência dos resultados, Intervalos de Confiança (IC) foram calculados para os resultados das rodadas de cada simulação, permitindo constatar o Nível de Confiança ( $\gamma$ ). Cada simulação executa, inicialmente,  $N$  passos até que se obtenha um  $\gamma \geq 90\%$ , com nível de significância de 10%. Este valor de  $N$  é, então, utilizado como limite em todas as outras rodadas de cada simulação no experimento.

Tabela 7.1: Parâmetros nas Simulações

Simulação	1	2	3	4	5	6	7	8	9	10
Datacenters	4	4	4	4	4	4	4	4	4	4
Hosts/DC	4	8	4	8	4	8	4	8	4	8
Núcleos/Host	4	4	4	4	4	4	4	4	4	4
Capacidade por núcleo (mips)	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
VMs	2	2	4	4	8	8	16	16	32	32
Workloads	2	2	4	4	8	8	16	16	32	32
Brokers	2	4	2	4	2	4	2	4	2	4
Núcleos/VM	1	1	1	1	1	1	1	1	1	1
Demanda de CPU por VM (mips)	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000

Integrar diferentes ações de gerenciamento pode tornar o custo de uma migração ainda mais alto, aumentando a possibilidade de inviabilização do uso da migração na alocação dinâmica de recursos. Por essa razão, duas dimensões são exploradas nos experimentos nesta tese: a efetividade da migração e a confirmação de que as escolhas que nortearam sua definição não causam um impacto que impossibilite sua adoção.

Dois experimentos são apresentados. No primeiro são coletadas a quantidade total de migrações que ocorre em ambos os esquemas (gerenciado e não gerenciado). O segundo experimento analisa a quantidade total de *hosts* sobrecarregados.

#### 7.4 EXPERIMENTO 1: TAXA DE SUCESSO EM MIGRAÇÕES

Neste experimento é demonstrado que o uso de ações de gerenciamento reduz o desperdício ocasionado por planejamentos de migrações mal sucedidas. Pode-se constatar que problemas como: indisponibilidade de um *host* de destino, informações desatualizadas da oferta de recursos e a falta de controle de VMs que têm suas cargas de trabalho concluídas, contribuem para o insucesso de uma migração. Ou seja, considera-se que há sucesso em uma migração se, de fato, a VM foi transferida para o *host* de destino escolhido, denotando, também o sucesso nas escolhas do planejamento.

Para analisar o quanto um cenário contribui para migrações bem sucedidas foi considerada a métrica designada como taxa de sucesso em migração ( $txSc$ ). Trata-se da relação entre o número de migrações bem sucedidas ( $nmbs$ ) e o número total de migrações que foram planejadas ( $nmt$ ), ou seja:

$$txSc = \frac{nmbs}{nmt} \quad (7.5)$$

Os valores de  $txSc$ , expressos em percentual, são descritos na Figura 7.1. Para cada simulação, são coletados  $txSc$  para cada rodada e, posteriormente, o valor médio de todos as

taxas determina o taxa de sucesso de uma simulação em particular ( $txScSim$ ). Ou seja,  $txScSim$  é dado por:

$$txScSim(x) = \frac{\sum_{i=1}^n txSc(i)}{n} \quad (7.6)$$

onde  $i$  identifica uma rodada particular,  $x$  uma simulação,  $n$  a quantidade de rodadas previamente definida dentro de um intervalo de confiança e  $txSc(i)$  a taxa de sucesso de migrações de uma rodada particular

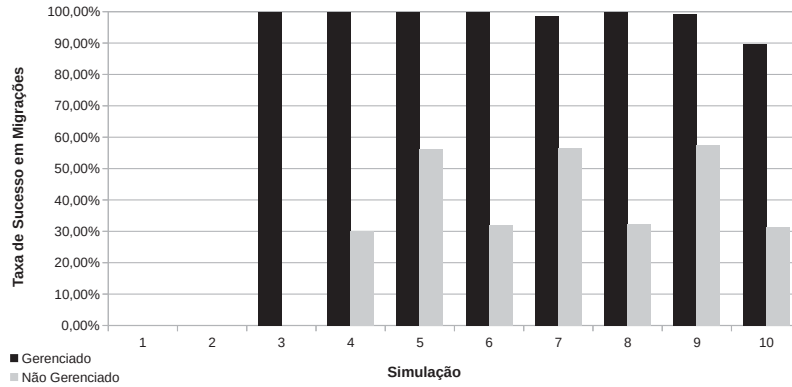


Figura 7.1: Taxa de sucesso em migrações em cada simulação do experimento

Na Figura 7.1 percebe-se que  $txScSim$  é 100% na maioria das simulações do cenário gerenciado. No entanto, os problemas que contribuem para migrações mal sucedidas são mais comuns em um cenário não gerenciado. Essa constatação pode ser observada na Figura 7.2, que coletou a quantidade média de migrações que não puderam ser concluídas, especificamente por conta da indisponibilidade do *host* de destino durante a migração.

A métrica adotada para a construção do gráfico da Figura 7.2 é designada  $txFail$ , ou seja, a razão entre a quantidade de indisponibilidade de *hosts* detectadas após o planejamento de migrações ( $nih$ ) em relação ao número de tentativas de migração ( $nmt$ ). Em cada rodada de uma simulação é calculada uma  $txFail$ , ou seja:

$$txFail(i) = \frac{nih(i)}{nmt(i)} \quad (7.7)$$

onde  $i$  identifica uma rodada particular.

Os valores identificados em cada simulação na Figura 7.2 é a média entre cada valor de  $txFail(i)$  identificado em cada simulação, portanto, tem-se:

$$txFailSim(x) = \frac{\sum_{i=1}^n txFail(i)}{n} \quad (7.8)$$

onde  $i$  identifica uma rodada particular,  $x$  uma simulação,  $n$  a quantidade de rodadas previamente definida dentro de um intervalo de confiança e  $txFailSim(i)$  a taxa de falhas no planejamento de migrações de uma rodada particular.

A falta de uma estratégia para reservas de recursos e o controle inexistente das migrações em curso são as principais motivações para a indisponibilidade de um *host* de destino. Pode-se também perceber que a adoção de políticas triviais para seleção de VMs candidatas a migrar e de *hosts* de destino também contribuíram para a ocorrência de insucesso nas migrações do cenário não gerenciado. De fato, percebeu-se que a adoção de uma política FCFS (*First Come First*

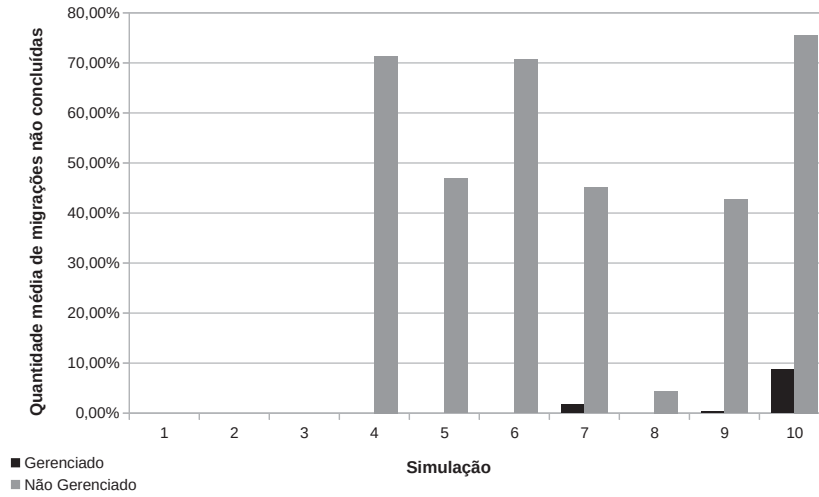


Figura 7.2: Taxa de falhas em migrações em cada simulação do experimento

*Served*) aliada aos dois problemas citados anteriormente conduziu, sistematicamente, a escolha de VMs e *hosts* que já estavam comprometidos em algum planejamento.

Neste experimento, portanto, pode-se concluir que a adoção de ações de gerenciamento levam a decisões de migração mais estáveis, ou seja, que têm maior chance de sucesso.

## 7.5 EXPERIMENTO 2: QUANTIDADE DE HOSTS SOBRECARGADOS

Neste experimento é analisada a capacidade de manter a oferta de recursos estável. Ou seja, se as migrações no cenário gerenciado produzem resultados que diminuam ou eliminem a ocorrência de episódios de instabilidade. Para analisar se esse propósito é atingido foram consideradas as sobrecargas em um *host*. Considera-se que um *host* está sobrecarregado quando o uso do processador estiver acima de um *threshold* estabelecido. Ou seja, em ambos os cenários, sempre que um *host* ultrapassa esse limite, uma sobrecarga é detectada.

O total de sobrecargas de um *host*  $i$  em uma simulação ( $totSob(h_i)$ ) é a quantidade total de sobrecargas que ocorrem em todas as rodadas de uma simulação, ou seja:

$$totSob(h_i) = \sum_{x=1}^n sob(round_x) \quad (7.9)$$

onde  $x$  indica uma das  $n$  rodadas de uma simulação.

A métrica quantidade média de sobrecargas ( $medSob$ ) é utilizada para permitir identificar o volume de sobrecargas em cada simulação. Para se identificar o valor de  $medSob$  foi computado o total de episódios de sobrecarga ocorrido em cada rodada ( $totSob$ ).  $medSob$  é a média do resultado de cada  $totSob$ , dada por:

$$medSob = (\sum_{i=1}^n totSob_i) / n \quad (7.10)$$

onde  $n$  é a quantidade de rodadas definidas com base em um intervalo de confiança aceitável.

A quantidade média de sobrecargas ocorreu em maior frequência em cenários não gerenciados, no entanto, na Figura 7.3 percebe-se que em algumas simulações, a quantidade de sobrecargas se aproxima em ambos os cenários. Foi possível constatar que a ausência de ações de gerenciamento tais como a reserva de recursos e seu controle durante o planejamento

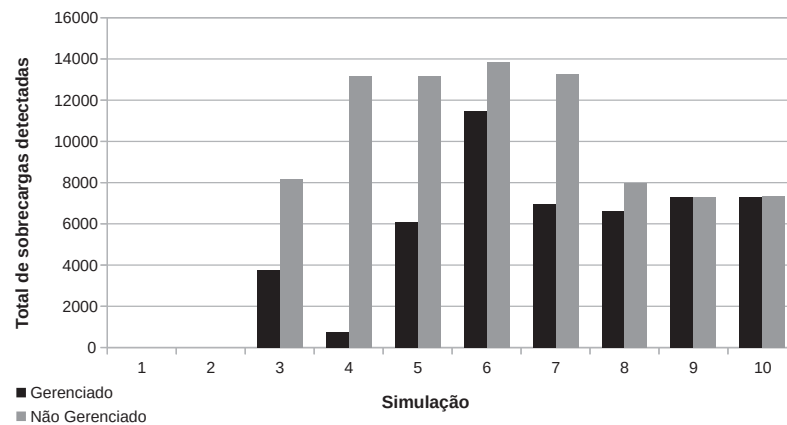


Figura 7.3: Total de sobrecargas de *hosts* em cada simulação

e a execução de migrações produz mais episódios de sobrecarga. De fato, a falta de reserva de recursos, por exemplo, estimula que *hosts* escolhidos como destino de migrações tornem-se sobrecarregados pela falta de controle de seus recursos. Essa situação pode ser constatada com a métrica resobrecarga, utilizada para identificar a quantidade de sobrecargas dos mesmos *hosts* que se repetem em cada simulação.

Considera-se uma resobrecarga o evento em que um *host* que já foi detectado sobrecarregado, torna-se novamente sobrecarregado. Neste contexto, pode-se considerar que quaisquer tentativas de reequilibrar o *host* não teve sucesso, denotando, portanto, que más escolhas conduziram o *host* a manter instável.

Na Figura 7.4 é apresentado o total de vezes em que o mesmo *host* tornou-se sobrecarregado em cada rodada de uma simulação. O resultado expresso é a soma de todos os episódios de sobrecarga em cada rodada de uma simulação.

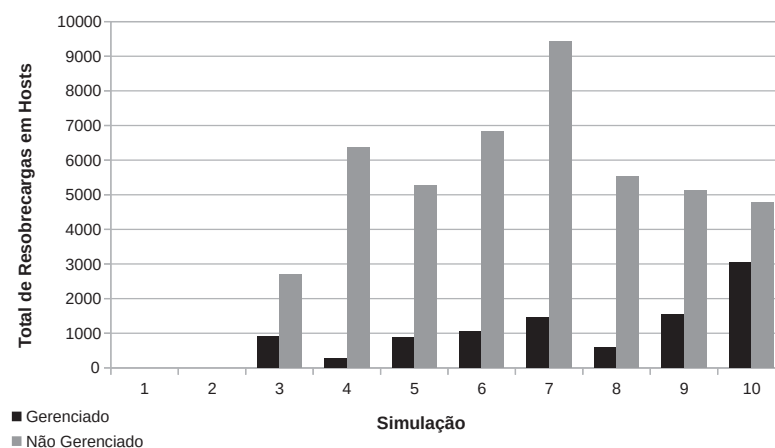


Figura 7.4: Total de sobrecargas que se repetem em cada simulação

As resobrecargas em maior quantidade no cenário não gerenciado podem explicar a maior instabilidade e a constância desses episódios, permitindo concluir que a falta de ações de gerenciamento conduzem a essa situação.

Outra métrica que também possibilita conclusões sobre a estabilidade das decisões tomadas durante o planejamento de uma migração é a remigração (*reMig*). Trata-se da quantidade de vezes que uma mesma VM precisa migrar novamente, denotando escolhas mal feitas para o

planejamento. Na Figura 7.5 é apresentada a quantidade média de remigrações do experimento (*AvgReMig*).

Para o cálculo de *AvgReMig* foram coletadas a quantidade total de remigrações de cada rodada  $i$  ( $reMig(i)$ ). Ou seja, *AvgReMig* é a média das quantidades totais identificadas em cada rodada, expressa como:

$$AvgReMig(x) = (\sum_{i=1}^n reMig(i))/n \quad (7.11)$$

onde  $i$  identifica uma rodada particular,  $x$  uma simulação,  $n$  a quantidade de rodadas previamente definida dentro de um intervalo de confiança e  $AvgReMig(x)$  a quantidade média de remigrações que ocorreram em uma simulação  $x$ .

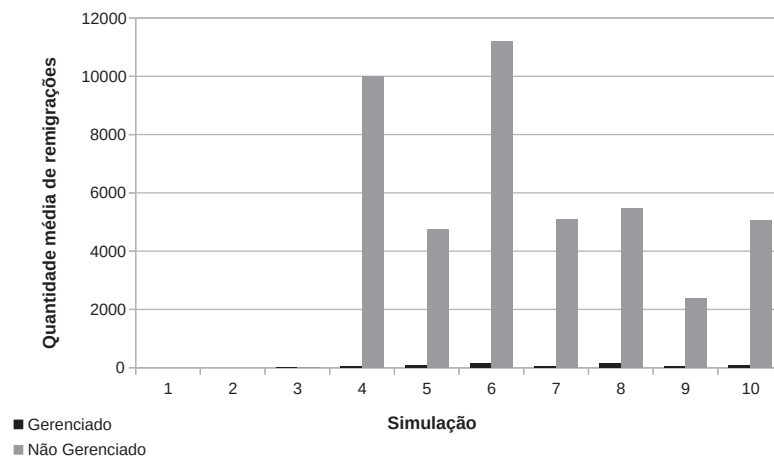


Figura 7.5: Quantidade média de migrações de VM que se repetem

As remigrações são mais constantes em um cenário não gerenciado, permitindo que se infira que a falta de ações de gerenciamento produzem planos de migração que não solucionam as instabilidades (sobrecarga) dos *hosts*. Ou seja, o não gerenciamento das migrações produz um panorama instável de oferta dos recursos, exigindo um esforço maior para a manutenção dos serviços em curso.

## 7.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada a avaliação da arquitetura proposta nesta tese. Optou-se por avaliar a proposta através de simulações, considerando o alto custo para se compor um ambiente de testes e também a flexibilidade que simulações oferecem ao permitir que detalhes finos sejam melhor investigados.

As simulações foram conduzidas em um ambiente baseado na ferramenta de simulação Cloudsim, que foi estendido para prever comportamentos não previstos originalmente, tais como, a integração de *Datacenters*, a migração de VMs, a criação sob demanda VMs e *workloads*, políticas específicas para seleção de VMs e *hosts* durante uma migração, políticas específicas para a alocação de VMs, *hosts* e *workloads*.

Para se obter resultados considerados confiáveis, cada experimento foi composto de 10 simulações. Cada simulação foi executada N vezes (ou em N rodadas). O valor adequado de N foi obtido a partir da identificação de um intervalo de confiança com nível de confiança de, no mínimo, 90%.

O primeiro experimento permitiu comprovar que a abordagem gerenciada proposta para a execução das simulações proveu resultados superiores no tocante à conclusão das migrações. Dessa forma, pode-se constatar que com a adoção de ações de gerenciamento, migrações tendem a ter maior taxa de sucesso em suas conclusões. Por outro lado, a falta de ações de gerenciamento demonstrou que as migrações tendem a ser mais frequentes, evidenciando que há limitações no planejamento das migrações.

Três aspectos puderam ser inferidos sobre as causas de um mau planejamento. O primeiro é a inexistência da reserva de recursos, aspecto que contribui para que migrações sejam interrompidas, causando desperdício no esforço de planejamento e gerando necessidades mais frequentes de migração assim como reiteradas sobrecargas de *hosts* que, por não conseguir ajustar sua oferta de recursos com uma migração, acabam por manter-se sobrecarregados.

O segundo experimento confirma o que foi anteriormente mencionado, contribuindo para se observasse uma maior e mais constante sobrecarga de *hosts* nas simulações baseados no cenário não gerenciado.

Dos experimentos é possível constatar que a adoção das ações de gerenciamento, conforme a arquitetura de gerenciamento de migrações proposto possibilita soluções mais eficazes de oferta de recursos assim como da manutenção dos serviços e infraestrutura subjacente.



## 8 CONCLUSÕES

Características associadas à virtualização tais como a provisão flexível de recursos, o isolamento e a migração de VMs são contribuições para um uso mais efetivo dos recursos em DCs de nuvens. Neste contexto, a migração de VMs tornou-se a estratégia *de facto* na provisão dinâmica de recursos em nuvem. No entanto, a constante variação na demanda por recursos exige que a tarefa de planejar migrações distancie-se da presença constante do operador humano. De fato, é comum que as alterações nos valores dos dados utilizados no planejamento de migrações não sejam prontamente identificadas pelos operadores humanos, contribuindo para que as migrações sejam mal planejadas e suas execuções produzam resultados ruins.

Pode-se constatar que a constante necessidade de intervenção humana deve-se à ausência de estratégias para controle e monitoramento dos dados. Em geral, os diferentes CMPs dependem de ferramentas externas para coleta de dados sobre as VMs. Sem a existência de interfaces que permitam a comunicação entre os diferentes componentes do ambiente, um operador humano precisa fazer esse controle.

Este trabalho propôs-se a estabelecer um conjunto de ações para promover o gerenciamento de recursos em um ambiente de nuvem, sendo designado como gerência de migrações de VMs. Cada etapa do ciclo de vida de uma migração é contemplada com ações de gerenciamento que contribuem para que os dados, de diferentes fontes, possam ser usados, permitindo a construção sob demanda de planos de migração. Na etapa inicial, a detecção de uma efetiva necessidade de migração ocorre com base na coleta das demandas de recursos das VMs de cada *host*. Para evitar decisões baseadas em eventos temporários, a análise de sobrecargas dos *hosts* é baseada em janelas deslizantes. Dois tamanhos de janelas são utilizadas: para a avaliação das condições dos *hosts* e para reavaliação do tamanho da janela.

Percebeu-se que a arquitetura que apoia a abordagem de gerenciamento devia contribuir para que no conjunto de dados utilizados no planejamento das migrações também fossem considerados os dados das migrações em curso. Dessa forma, os dados da execução das migrações também foi considerada durante o planejamento.

Para o planejamento foi considerada uma política para seleção das VMs selecionadas para migrar. Decidiu-se por organizar a lista de VMs candidatas a migrar conforme a relação quantidade de dados a migrar e o tempo gasto para a migração. Nesse caso, as VMs com maiores valores para a referida relação eram escolhidas. A escolha dos *hosts* que hospedariam as VMs migrantes também baseou-se em uma política de seleção que valoriza apenas os *hosts* que podem atender à demanda esperada da VM migrante.

A arquitetura foi avaliada em dois experimentos, baseados na comparação com um cenário desprovido de ações de gerenciamento. Ou seja, que não possui suporte à reserva de recursos durante uma migração; que não atesta se a violação de *thresholds* decorre de um evento temporário; que não considera os dados relativos às migrações em curso e que baseiam-se em políticas elementares para seleção de VMs candidatas a migrar e novos *hosts* de destino (ou seja, baseiam-se em uma política FCFS (*First Come First Served*)).

No primeiro experimento avaliou-se a taxa de sucesso das migrações, uma medida que decorre da relação entre a quantidade de migrações concluídas e as que foram tentadas. Observou-se no cenário não gerenciado a taxa de sucesso foi inferior em todos os casos, tendo apresentado uma quantidade de falhas sempre maior. Constatou-se que a ausência de uma política de seleção de VMs e de *hosts* de destino contribuiu para, no cenário não gerenciado,

gerar escolhas de VMs e *hosts* já comprometidos, gerando, portanto, planos de migração mal sucedidos.

No segundo experimento analisou-se a quantidade de sobrecargas que ocorreram ao longo das simulações do experimento. No cenário não gerenciado a ocorrência de sobrecargas foi constantemente maior. Da mesma forma, o número de *hosts* que estiveram sobrecarregados mais de uma vez em cada simulação (resobrecarga). Ambos os resultados denotam que as instabilidades nos *hosts* mantêm-se sem solução em um cenário não gerenciado. Essa constância de instabilidades também pode ser observada na avaliação da quantidade de remigrações, ou seja, da quantidade de VMs que precisam migrar mais de uma vez em cada rodada. Em um cenário não gerenciado, as remigrações foram mais constantes e em maior número, permitindo concluir que a ausência de ações de gerenciamento foram vitais para o insucesso.

Além dos resultados identificados nos experimentos, deve-se ressaltar dois trabalhos que foram aceitos em conferências, cujas revisões foram cruciais para a melhoria do trabalho de tese. Destacamos as seguintes publicações:

- A. V. de Matos, C. Maziero. **An Architecture for the Management of Virtual Machine Migrations.** Artigo completo aceito na *14th IEEE International Systems Conference* (IEEE SysCon 2020), Qualis A-4.
- A. V. de Matos, C. Maziero. **Uma Arquitetura para a Gerência de Migração de Máquinas Virtuais.** Artigo completo aceito no XVIII Workshop em Clouds e Aplicações (WCGA) do Simpósio Brasileiro de Redes de Computadores (WCGA/SBRC 2020), Qualis B-4.

Há aspectos nesse trabalho que podem ser melhor explorados, sendo objeto de pesquisas futuras. Como o modelo de migrações apresentado nesta tese foca em um cenário inter-DC, no entanto, dentro do ambiente de uma nuvem, considera-se que explorar outro cenário baseado em múltiplas nuvens seja importante. A criação de um modelo de migrações para um cenário inter-DC e inter-nuvem e sua respectiva avaliação são úteis para analisar modelos de integração de nuvens, tais como uma federação de nuvens.

Foi possível constatar na execução deste trabalho que estratégias para alocação de VMs em *hosts* contribuem para que migrações sejam necessárias. Sendo a alocação parte do processo de gerenciamento de recursos, espera-se no futuro que seja conduzida uma avaliação do impacto das estratégias de alocação em métricas como a quantidade de remigrações e de sobrecargas de *hosts*.

Outro assunto que deverá ser tratado em um trabalho futuro é o delineamento de um modelo que permita gerenciar a capacidade ociosa dos recursos em decorrência do encerramento dos *workloads*. Este modelo deve ser integrado a um serviço de *brokering*, permitindo que novas cargas de trabalho possam ser requisitadas do ambiente, contudo, aproveitando a capacidade que ficou ociosa. Pretende-se que este modelo seja integrado à proposta de alocação de VMs considerada no segundo trabalho futuro.

## REFERÊNCIAS

- A. Elhabbash, F. A., Samreen, F., Hadley, J. e Elkhatib, Y. (2019). Cloud brokerage: A systematic survey. *ACM Computing Surveys (CSUR)*, 51(6):1–28.
- Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A. e Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications*, 52:11–25.
- Aldossary, M. e Djemame, K. (2018). Performance and energy-based cost prediction of virtual machines auto-scaling in clouds. Em *44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, páginas 502–509, Prague - Czech Republic.
- Alicherry, M. e Lakshman, T. V. (2012). Network aware resource allocation in distributed clouds. Em *31st Annual IEEE International Conference on Computer Communications (Infocom - 2012)*, páginas 963–971, Orlando - USA.
- Ayoub, O., Huamani, O., Musumeci, F. e Tornatore, M. (2019). Efficient online virtual machines migration for alert-based disaster resilience. Em *15th International Conference on the Design of Reliable Communication Networks (DRCN)*, páginas 146–153, Coimbra - Portugal.
- Baccarelli, E., Amendola, D. e Cordeschi, N. (2015). Minimum-energy bandwidth management for qos live migration of virtual machines. *Computer Networks*, 93:1–22.
- Basu, D., Wang, X., Hong, Y., Chen, H. e Bressan, S. (2019). Learn-as-you-go with megh: Efficient live migration of virtual machines. *IEEE Transactions on Parallel and Distributed Systems*, 30(8):1786–1801.
- Beloglazov, A. e Buyya, R. (2010). Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. Em *8th International Workshop on Middleware for Grids, Clouds and e-Science*, páginas 1–6, Bangalore - India.
- Beloglazov, A. e Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13):1397–1420.
- Beloglazov, A. e Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems*, 24(7):1366–1379.
- Bernstein, D. e Demchenko, Y. (2013). The ieee intercloud testbed—creating the global cloud of clouds. Em *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom - 2013)*, volume 2, páginas 45–50, Bristol - UK.
- Biran, O., Breitgand, D., Lorenz, D., Masin, M., Raichstein, E., Weit, A. e Iyoob, I. (2018). Heterogeneous resource reservation. Em *IEEE 10th International Conference on Cloud Engineering (IC2E - 2018)*, páginas 141–147, Nicosia - Cyprus.
- Bittencourt, L. F., Madeira, E. e Fonseca, N. (2015). Resource management and scheduling. *Cloud Services, Networking, and Management*, páginas 243–267.

- Boutaba, R., Zhang, Q. e Zhani, M. F. (2013). Virtual machine migration in cloud computing environments: Benefits, challenges, and approaches. *Communication Infrastructures for Cloud Computing*, páginas 383–408.
- Buyya, R., Ranjan, R. e Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. Em *International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP - 2010)*, páginas 13–31, New York - USA.
- Calheiros, R. N., Ranjan, R., Beloglazov, A., Rose, C. A. F. D. e Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, 41(1):23–50.
- Calheiros, R. N., Toosi, A. N., Vecchiola, C. e Buyya, R. (2012). A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems*, 28(8):1350–1362.
- Cao, R., Tang, Z., Li, K. e Li, K. (2018). Hmgowm: a hybrid decision mechanism for automating migration of virtual machines. *IEEE Transactions on Services Computing*.
- Carlini, E., Coppola, M., Dazzi, P., Ricci, L. e Righetti, G. (2011). Cloud federations in contrail. Em *European Conference on Parallel Processing*, páginas 159–168, Bordeaux - France.
- Chaabouni, T. e Khemakhem, M. (2018). Energy management strategy in cloud computing: a perspective study. *The Journal of Supercomputing*, 74(12):6569–6597.
- Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. e Warfield, A. (2005). Live migration of virtual machines. Em *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI 2007)*, páginas 273–286, Berkeley - USA.
- Council, C. S. (2017). Practical guide to cloud management platforms. <https://www.omg.org/cloud/deliverables/CSCC-Practical-Guide-to-Cloud-Management-Platforms.pdf>. Acessado em 22/01/2020.
- Cusumano, M. (2010). Cloud computing and saas as new computing platforms. *Communications of the ACM*, 53(4):27–29.
- Dargie, W. (2014). Estimation of the cost of vm migration. Em *23rd International Conference on Computer Communication and Networks (ICCCN - 2014)*, páginas 1–8, Shanghai - China.
- Durkee, D. (2010). Why cloud computing will never be free. *Queue*, 8(4):20.
- Elmroth, E. e Larsson, L. (2009). Interfaces for placement, migration, and monitoring of virtual machines in federated clouds. Em *Eighth International Conference on Grid and Cooperative Computing*, páginas 253–260, Lanzhou - China.
- Freet, D., Agrawal, R., Walker, J. J. e Badr, Y. (2016). Open source cloud management platforms and hypervisor technologies: A review and comparison. Em *IEEE SoutheastCon 2016*, páginas 1–8, Norfolk - USA.
- Galante, G. e de Bona, L. C. E. (2012). A survey on cloud computing elasticity. Em *IEEE Fifth International Conference on Utility and Cloud Computing (UCC -2012)*, páginas 263–270, Chicago - USA.

- Gilesh, M. P., Jain, S., Kumar, S. D. M., Jacob, L. e Bellur, U. (2020). Opportunistic live migration of virtual machines. *Concurrency and Computation: Practice and Experience*, 32(5):e5477.
- Gmach, D., Rolia, J., Cherkasova, L. e Kemper, A. (2009). Resource pool management: Reactive versus proactive or let's be friends. *Computer Networks*, 53(17):2905–2922.
- Goumas, G., Karakostas, V., Nikas, K., Siakavaras, D., Psomadakis, S., Gerangelos, S., Bayuh, L. e P. Svård, P. S. K. (2018). ACTiCLOUD: Activating resource efficiency and large databases in the cloud - deliverable 2.2. Relatório Técnico European Union's Horizon No. 732366, ACTiCLOUD Consortium.
- Goumas, G., Nikas, K., Lakew, E., Ewnetu, B., Kotselidis, C., Attwood, A., Elmroth, E., Flouris, M., Foutris, N., Goodacre, J. e Grohmann, D. (2017). ACTiCLOUD: Enabling the next generation of cloud applications. Em *IEEE 37th International Conference on Distributed Computing Systems (ICDCS - 2017)*, páginas 1836–1845, Atlanta - USA.
- Group, N. C. C. S. W. et al. (2013). Nist cloud computing security reference architecture. Relatório Técnico NIST SP 500-291, National Institute of Standards and Technology, Washington - USA.
- Han, S., Egi, N., Panda, A., Ratnasamy, S., Shi, G. e Shenker, S. (2013). Network support for resource disaggregation in next-generation datacenters. Em *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, páginas 1–7, College Park, USA.
- Herbst, N. R., Kounev, S. e Reussner, R. H. (2013). Elasticity in cloud computing: What it is, and what it is not. Em *10th International Conference on Autonomic Computing (ICAC - 2013)*, páginas 23–27, San Jose - USA.
- Hines, M. R. e Gopalan, K. (2009). Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. Em *Proceedings of the International Conference on Virtual Execution Environments (ACM SIGPLAN/SIGOPS - 2009)*, páginas 51–60, Washington - USA.
- Jegou, Y., Harsh, P., Cascella, R. G., Dudouet, F. e Morin, C. (2012). Managing ovf applications under sla constraints on contrail virtual execution platform. Em *8th International Conference on Network and service management (CNSM - 2012)*, páginas 399–405, Las Vegas - USA.
- Jennings, B. e Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, 23(3):567–619.
- Kapil, D., Pilli, E. S. e Joshi, R. C. (2013). Live virtual machine migration techniques: Survey and research challenges. Em *IEEE 3rd International Advance Computing Conference (IACC - 2013)*, páginas 963–969, Ghaziabad - India.
- Kollberg, S., Lakew, E. B., Svård, P., Elmroth, E. e J.Tordsson (2020). Spreading the heat: Multi-cloud controller for failover and cross-site offloading. Em *34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, páginas 1154–1164, Caserta - Italy.
- K.Ye, Jiang, X., Huang, D., Chen, J. e Wang, B. (2011). Live migration of multiple virtual machines with resource reservation in cloud computing environments. Em *4th International Conference on Cloud Computing*, páginas 267–274, Victoria - Australia.



- Liu, X., Wu, J., Sha, G. e Liu, S. (2020). Virtual machine consolidation with minimization of migration thrashing for cloud data centers. *Mathematical Problems in Engineering*, 2020.
- Luiz, S. O. D., Perkusich, A. e Lima, A. M. N. (2010). Multisize sliding window in workload estimation for dynamic power management. *IEEE Transactions on Computers*, 59(12):1625–1639.
- Mainardi, M. e Waite, A. (2018). Evaluation criteria for cloud management platforms and tools. <https://www.gartner.com/doc/3873016>. Acessado em 26/01/2020.
- Mandal, U., Chowdhury, P., Tornatore, M., Martel, C. e Mukherjee, B. (2016). Bandwidth provisioning for virtual machine migration in cloud: Strategy and application. *IEEE Transactions on Cloud Computing*, 6(4):967–976.
- Marinescu, D. C. (2013). *Chapter 6 - Cloud Resource Management and Scheduling*, páginas 163 – 203. Morgan Kaufmann.
- Mashtizadeh, A. L., Cai, M., Tarasuk-Levin, G., Koller, R., Garfinkel, T. e Setty, S. (2014). Xvmotion: Unified virtual machine migration over long distance. Em *Annual Technical Conference ({USENIX}{ATC} 2014)*, páginas 97–108, San Diego - USA.
- Medina, V. e García, J. M. (2014). A survey of migration mechanisms of virtual machines. *ACM Computing Surveys (CSUR)*, 46(3):1–33.
- Mell, P. e Grance, T. (2011). The nist definition of cloud computing. Relatório técnico, National Institute of Standards and Technology - Computer Security Division, Information Technology Laboratory,, Washington - USA.
- Mishra, M., Das, A., Kulkarni, P. e Sahoo, A. (2012). Dynamic resource management using virtual machine migrations. *IEEE Communications Magazine*, 50(9).
- Mishra, M. e Sahoo, A. (2011). On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach. Em *IEEE International Conference on Cloud Computing (CLOUD - 2011)*, páginas 275–282, Athens - Greece.
- Moreno-Vozmediano, R., Montero, R. e Llorente, I. M. (2012). IaaS cloud architecture: From virtualized datacenters to federated cloud infrastructures. *IEEE Computer*, 45(12):65–72.
- Parikh, S. M. (2013). A survey on cloud computing resource allocation techniques. Em *International Conference on Engineering*, páginas 1–5, Seoul - Korea.
- Pietri, I. e Sakellariou, R. (2016). Mapping virtual machines onto physical machines in cloud computing: A survey. *ACM Computing Surveys (CSUR)*, 49(3):49.
- Ponraj, A. (2019). Optimistic virtual machine placement in cloud data centers using queuing approach. *Future Generation Computer Systems*, 93:338–344.
- Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E. e Caceres, J. (2009a). Reservoir: Management technologies and requirements for next generation service oriented infrastructures. Em *IFIP/IEEE International Symposium on Integrated Network Management (IM - 2009)*, páginas 307–310, Long Island - USA.



- Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E. e Caceres, J. (2009b). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):4–1.
- Rosenblum, M. e Garfinkel, T. (2005). Virtual machine monitors: Current technology and future trends. *IEEE Computer*, 38(5):39–47.
- Sapuntzakis, C. P., Chandra, R., Pfaff, B., Chow, J., Lam, M. S. e Rosenblum, M. (2002). Optimizing the migration of virtual computers. *ACM SIGOPS Operating Systems Review*, 36(SI):377–390.
- Sharma, S. e Chawla, M. (2013). A technical review for efficient virtual machine migration. Em *International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, páginas 20–25, Pune - India.
- Smith, J. e Nair, R. (2005). *Virtual machines: versatile platforms for systems and processes*. Elsevier.
- Strunk, A. (2012). Costs of virtual machine live migration: A survey. Em *IEEE Eighth World Congress on Services*, páginas 323–329, Honolulu - USA.
- Svard, P., Hudzia, B., Walsh, S., Tordsson, J. e Elmroth, E. (2014). The noble art of live vm migration - principles and performance of pre copy, post copy and hybrid migration of demanding workloads. Relatório Técnico UMINF-12.11, UmeåUniversity, Umeå- Sweden.
- Takahashi, T., Blanc, G., Kadobayashi, Y., Fall, D., Hazeyama, H. e Matsuo, S. (2012). Enabling secure multitenancy in cloud computing: Challenges and approaches. Em *2nd Baltic Congress on Future Internet Communications*, páginas 72–79, Vilnius - Lithuania.
- Tani, H. G. e Amrani, C. E. (2016). Cloud computing CPU allocation and scheduling algorithms using CloudSim simulator. *Internationall Journal of Electrical and Computer Engineering*, 6(4):2088–8708.
- Tordsson, J., Montero, R. S., Moreno-Vozmediano, R. e Llorente, I. M. (2012). Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future generation computer systems*, 28(2):358–367.
- Vogels, W. (2008). Beyond server consolidation. *Queue*, 6(1):20–26.
- Voorsluys, W., Broberg, J., Venugopal, S. e Buyya, R. (2009). Cost of virtual machine live migration in clouds: A performance evaluation. Em *IEEE International Conference on Cloud Computing*, páginas 254–265, Lanzhou - China.
- Wickboldt, J. A., Esteves, R. P., de M. B. Carvalho e Granville, L. Z. (2014). Resource management in iaas cloud platforms made flexible through programmability. *Computer Networks*, 68:54–70.
- Wood, T., Shenoy, P. J., Venkataramani, A. e Yousif, M. S. (2007). Black-box and gray-box strategies for virtual machine migration. Em *4th Symposium on Networked Systems Design and Implementation (NSDI 2007)*, volume 7, páginas 17–28, Cambridge - USA.

- Woudenberg, S. V. (2016). Lessons learned from a year of using live migration in production on google cloud. <https://cloud.google.com/blog/products/gcp/lessons-learned-from-a-year-of-using-live-migration-in-production-on-google-cloud>. Acessado em 08/02/2020.
- Xu, J. e Fortes, J. (2011). A multi-objective approach to virtual machine management in datacenters. Em *Proceedings of the 8th ACM international conference on Autonomic computing*, páginas 225–234, Karlsruhe - Germany.
- Zhang, F. (2018). *Challenges and New Solutions for Live Migration of Virtual Machines in CloudComputing Environments*. Tese de doutorado, Georg-August University School of Science, Göttingen - Alemanha. 180 pgs.
- Zhang, F., Fu, X. e Yahyapour, R. (2018a). Layermover: Fast virtual machine migration over wan with three-layer image structure. *Future Generation Computer Systems*, 83:37–49.
- Zhang, F., Liu, G., Fu, X. e Yahyapour, R. (2018b). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys and Tutorials*, 20(2):1206–1243.
- Zhang, W., Han, S., He, H. e Chen, H. (2017). Network-aware virtual machine migration in an overcommitted cloud. *Future Generation Computer Systems*, 76:428–442.
- Zhao, H., Wang, Q., Wang, J., Qi, G., Yang, P. e Qiao, L. (2019). A performance-guaranteed virtual machine migration strategy in cloud. Em *IEEE 21st International Conference on High Performance Computing and Communications (HPCC-2019)*, páginas 1894–1899, Zhangjiajie - China.
- Zheng, W., Feng, C., Yu, T., Yang, X. e Wu, X. (2019). Towards understanding bugs in an open source cloud management stack: An empirical study of openstack software bugs. *Journal of Systems and Software*, 151:210–223.